



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Control y monitorización de una finca con IoT y LPWAN

Autor/es

ALEJANDRO NAVARRO LORENTE

Director/es

CARLOS ELVIRA IZURRATEGUI y SILVANO NÁJERA CANAL ,

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2018-19



Control y monitorización de una finca con IoT y LPWAN, de ALEJANDRO NAVARRO LORENTE

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor, 2019

© Universidad de La Rioja, 2019

publicaciones.unirioja.es

E-mail: publicaciones@unirioja.es



**UNIVERSIDAD
DE LA RIOJA**

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE GRADO

**TITULACIÓN: Grado en
Ingeniería Electrónica Industrial y Automática**

CURSO: 2019/2020 CONVOCATORIA: SEPTIEMBRE

TÍTULO:

Control y monitorización de una finca con IoT y LPWAN

AUTOR: Alejandro Navarro Lorente

**DIRECTOR/ES: Carlos Elvira Izurrategui
Silvano Nájera Canal**

DEPARTAMENTO: Ingeniería Eléctrica

 ÍNDICE

ÍNDICE.....	1
ABSTRACT	4
RESÚMEN DOCUMENTAL.....	5
RESÚMEN EXTENDIDO	6
CAPÍTULO 1. MEMORIA.....	7
ÍNDICE MEMORIA.....	7
ÍNDICE DE ILUSTRACIONES.....	9
ÍNDICE DE TABLAS	11
1.1. Objeto.....	12
1.2. Alcance	14
1.3. Antecedentes	17
1.4. Normas y referencias	18
1.4.1. Disposiciones legales y normas aplicadas	18
1.4.2. Programas de cálculo	18
1.4.3. Plan de gestión de la calidad aplicado durante la redacción del Proyecto	20
1.4.4. Bibliografía	23
1.4.5. Otras referencias	25
1.5. Definiciones y abreviaturas.....	26
1.6. Requisitos de diseño	28
1.6.1. El cliente	28
1.6.2. Emplazamiento y entorno socioeconómico y ambiental	28
1.6.3. Estudios encaminados a la definición de la solución adoptada.....	28
1.7. Análisis de soluciones.....	34
1.7.1. Elección de las comunicaciones y el hardware necesario.....	34
1.7.2. Elección del servidor	35
1.7.3. Elección de la interfaz de usuario.....	37
1.8. Resultados finales.....	38
1.8.1. Modificación de la instalación eléctrica para el control y la monitorización del sistema	38
1.8.2. Comunicación LoRa punto a punto	45
1.8.3. Software de control para ESP32.....	52

1.8.4.	Servidor propio ubicado en Raspberry Pi.....	59
1.8.5.	Aplicación móvil para Android	72
1.8.6.	Ideas para mejoras en futuras actualizaciones	80
1.9.	Planificación	83
1.10.	Orden de prioridad entre los documentos.....	86
CAPÍTULO 2. ANEXOS		87
ÍNDICE DE ANEXOS		87
ÍNDICE DE ILUSTRACIONES.....		90
2.1.	Documentación de partida.....	91
2.2.	Cálculos	92
2.3.	Estudio de tecnologías LPWAN	94
2.3.1.	Sigfox	94
2.3.2.	LoRa.....	95
2.3.3.	LoRaWAN.....	96
2.4.	Prueba de comunicación real entre la finca y la vivienda	97
2.5.	Prueba de comunicación real con el inversor	98
2.6.	Modelo de simulación	99
2.6.1.	Maqueta	99
2.6.2.	Software simulador del inversor	99
2.7.	Código desarrollado	102
2.7.1.	Android Studio.....	102
2.7.2.	ESP32.....	206
2.7.3.	PHP	228
CAPÍTULO 3. PLANOS		241
ÍNDICE DE PLANOS		241
3.1.	Esquema WiFi LoRa 32 (V2).....	242
3.2.	Esquema de control	243
3.3.	Esquema de potencia.....	244
CAPÍTULO 4. PLIEGO DE CONDICIONES.....		245
ÍNDICE PLIEGO DE CONDICIONES.....		245
ÍNDICE DE ILUSTRACIONES.....		246
ÍNDICE DE TABLAS.....		246
4.1.	Objeto del pliego de condiciones.....	247
4.2.	Condiciones técnicas.....	247
4.2.1.	Descripción de las nuevas instalaciones	247

4.2.2.	Especificaciones de los elementos constitutivos del proyecto	250
4.2.3.	Condiciones y comprobación de funcionamiento.....	255
4.3.	Ejecución	258
4.4.	Condiciones económicas	258
4.4.1.	Garantía de funcionamiento	258
4.4.2.	Impuestos	258
4.4.3.	Actualización de precios.....	259
4.4.4.	Reclamación por un aumento de precios	259
4.4.5.	Retraso en los pagos	259
4.4.6.	Retraso en los trabajos.....	259
4.5.	Condiciones legales	260
4.5.1.	Marco jurídico	260
4.5.2.	Accidentes y daños a terceros.....	260
4.5.3.	Responsabilidad civil	260
4.5.4.	Permisos	260
4.5.5.	Rescisión del contrato	261
4.5.6.	Normas y legislación aplicadas.....	261
CAPÍTULO 5. MEDICIONES.....		263
ÍNDICE DE MEDICIONES		263
5.1.	Elementos materiales.....	264
5.2.	Mano de obra	267
CAPÍTULO 6. PRESUPUESTO		268
ÍNDICE DE PRESUPUESTO.....		268
6.1.	Cuadro de precios unitarios	269
6.1.1.	Elementos materiales.....	269
6.1.2.	Mano de obra	270
6.2.	Cuadro de precios unitarios de las unidades de obra	271
6.3.	Presupuesto	273
6.3.1.	Elementos materiales.....	273
6.3.2.	Mano de obra	274
6.3.3.	Impuestos y tasas	274
6.3.4.	Presupuesto.....	275

ABSTRACT

In this document an *IoT* system with *Lora* point-to-point communication has been developed for data exchange. Moreover, it has been implemented its own server for data storage and an Android application for the user interface.

This system has been applied to a private rural property with an independent solar energy installation, a swimming pool and an orchard. Some of the new functionalities given are: monitoring the solar energy production, control the orchard irrigation, control the maintenance of the swimming pool, etc. *LPWAN LoRa* technology has been studied and applied, allowing that a network of sensors and actuators can connect remotely over a significantly wide distance. In this way it has been managed to provide connectivity to the system at a point without internet access. After the studies and tests carried out, results were satisfactory, obtaining a connection between the two *LoRa* communication points from 2km away. A new *LoRa* protocol has been designed in order to optimise the size of the data sending between both points. The own server implemented on a Raspberry Pi has been achieved, at a very low cost, to store the generated data and work as a link between the Android application and the system. In addition, the API has been designed considering the system, so it is fully integrated. The entire installation has been developed independently of external services, except of the internet connection, which already existed.

In conclusion, a total operation of the *IoT* application has been achieved using *LoRa*, although there are still some points of improvement and optimization.

RESÚMEN DOCUMENTAL

En el presente documento se ha realizado una implementación de un sistema *IoT* con una comunicación *LoRa* punto a punto para el intercambio de datos, así como el desarrollo de un servidor propio para el almacenamiento de los datos y una aplicación Android para la interfaz de usuario.

Se ha aplicado este sistema a una finca rústica particular con una instalación solar fotovoltaica, una piscina y una huerta. Con la nueva instalación se han añadido funcionalidades como la monitorización de la producción de energía y el control remoto y automatizado de riego y mantenimiento de la piscina. Se ha estudiado y aplicado la tecnología *Low Power Wide Area Network (LPWAN) LoRa* que permite conectar una red de sensores y actuadores de forma remota a una distancia considerablemente amplia. De este modo se ha conseguido dar conectividad al sistema en un punto sin acceso a internet. Tras los estudios y pruebas realizados se han obtenido resultados satisfactorios, obteniendo una conexión entre los dos puntos de comunicación *LoRa* de 2km de distancia. Se ha desarrollado un protocolo de comunicación sobre la conexión *LoRa* para optimizar el envío de información en el menor número de bytes posible. El servidor propio implementado sobre una Raspberry Pi ha conseguido, a muy bajo coste, almacenar los datos generados y ser un nexo de unión entre la aplicación Android y el sistema. Además, la *API* ha sido diseñada teniendo en cuenta la todo el sistema, por lo que está completamente integrada. La totalidad de la instalación se ha desarrollado independiente de servicios externos, exceptuando la conexión a internet, que ya existía previamente.

En conclusión, se ha conseguido un funcionamiento total de la aplicación *IoT* con la utilización de *LoRa*, a pesar de que todavía existen puntos de mejora y optimización.

RESÚMEN EXTENDIDO

El proyecto descrito a continuación desarrolla una implementación de *IoT* desde la toma de datos en sensores, hasta la interfaz de usuario, incluyendo un servidor propio para el almacenamiento de los datos.

Durante el desarrollo del proyecto se han creado todas las partes necesarias del sistema para poder llevar a cabo las tareas requeridas. Algunas de estas tareas son: monitorización de una instalación solar de autoconsumo independiente, control de riego, control de mantenimiento de una piscina, etc.

Para poder llevar a cabo estas tareas ha sido necesario desarrollar varias partes que trabajen en conjunto para lograr los objetivos.

En primer lugar, se desarrolló un sistema de control bajo un ESP32, en el entorno de Arduino, que permite controlar una serie de actuadores digitales, leer determinados sensores, y monitorizar los datos de la instalación fotovoltaica que ofrece el inversor vía puerto serie.

Para poder intercambiar datos desde la interfaz de usuario hasta el controlador, se utilizó tecnología *Low Power Wide Area Network (LPWAN)*, en concreto una comunicación *LoRa* punto a punto, debido a que no se dispone de conexión a internet en la finca.

Los datos son enviados hasta una vivienda a 2km de la finca donde existe conexión a internet y desde esta vivienda se pueden guardar los datos en un servidor.

El servidor utilizado para guardar los datos también fue desarrollado durante la elaboración de este proyecto. Para ello se utilizó un ordenador de bajas prestaciones, pero suficiente para el limitado número de usuarios y datos que va a utilizar el sistema. En concreto se montó un servidor tipo *LAMP* (Linux, Apache, MySQL y PHP) en una Raspberry Pi modelo 3B+. Una vez montado el servidor se generaron las tablas necesarias de la base de datos, y se desarrolló una API pensada para interactuar tanto con el ESP32 como con el sistema operativo Android.

Teniendo ya un sistema para intercambiar datos con la monitorización y el control de la finca, únicamente quedaba desarrollar la interfaz de usuario. Como ya se ha mencionado se utilizó Android para tal cometido. Se utilizó el entorno de desarrollo que proporciona Google de manera gratuita, Android Studio. La interfaz de usuario permite monitorizar el estado del sistema, así como interactuar con los actuadores y programar acciones automáticas.

En conclusión, se ha desarrollado un sistema completo, desde el control hasta la interfaz de usuario, incluyendo el servidor para almacenar los datos, como solución a este proyecto.

 CAPÍTULO 1. MEMORIA

ÍNDICE MEMORIA

1.1.	Objeto.....	12
1.2.	Alcance	14
1.3.	Antecedentes	17
1.4.	Normas y referencias	18
1.4.1.	Disposiciones legales y normas aplicadas	18
1.4.2.	Programas de cálculo	18
1.4.3.	Plan de gestión de la calidad aplicado durante la redacción del Proyecto	20
1.4.4.	Bibliografía	23
1.4.4.1.	Información sobre LoRa	23
1.4.4.2.	ESP32, Arduino y librerías LoRa.....	23
1.4.4.3.	Información sobre el inversor presente en la instalación.....	23
1.4.4.4.	Raspberry Pi y servidor.....	24
1.4.4.5.	Android Studio.....	24
1.4.5.	Otras referencias.....	25
1.5.	Definiciones y abreviaturas.....	26
1.6.	Requisitos de diseño	28
1.6.1.	El cliente	28
1.6.2.	Emplazamiento y entorno socioeconómico y ambiental.....	28
1.6.3.	Estudios encaminados a la definición de la solución adoptada.....	28
1.6.3.1.	Estudio de alternativas para la comunicación con Internet.....	28
1.6.3.2.	Estudio de las alternativas del almacenamiento de datos en la nube.....	30
1.6.3.3.	Alternativas de la interfaz de usuario	32
1.7.	Análisis de soluciones.....	34
1.7.1.	Elección de las comunicaciones y el hardware necesario.....	34
1.7.2.	Elección del servidor	35
1.7.3.	Elección de la interfaz de usuario.....	37
1.8.	Resultados finales.....	38
1.8.1.	Modificación de la instalación eléctrica para el control y la monitorización del sistema	38

1.8.2.	Comunicación LoRa punto a punto	45
1.8.2.1.	Configuración del módulo	46
1.8.2.2.	Protocolo desarrollado.....	47
1.8.3.	Software de control para ESP32.....	52
1.8.3.1.	Software en el módulo de la finca.....	52
1.8.3.2.	Software en el punto de conexión a internet	56
1.8.4.	Servidor propio ubicado en Raspberry Pi.....	59
1.8.4.1.	Tipo de servidor y proceso de instalación.....	59
1.8.4.2.	Tablas de la base de datos	64
1.8.4.3.	API de interacción con el servidor.....	67
1.8.5.	Aplicación móvil para Android	72
1.8.5.1.	Entorno gráfico de la interfaz de usuario.....	72
1.8.5.2.	Conexión con el servidor y funcionamiento de la aplicación.....	79
1.8.6.	Ideas para mejoras en futuras actualizaciones	80
1.8.6.1.	Servidor redundante	80
1.8.6.2.	Nuevas Interfaces de usuario.....	80
1.8.6.3.	Copias de seguridad automáticas	80
1.8.6.4.	Mejoras de la gestión de la base de datos	81
1.8.6.5.	Mejoras de la aplicación Android.....	81
1.8.6.6.	Mejoras del programa de control	81
1.8.6.7.	Mejor gestión del consumo de agua.....	82
1.9.	Planificación	83
1.10.	Orden de prioridad entre los documentos.....	86

ÍNDICE DE ILUSTRACIONES

Ilustración 1. 1. Placas solares instaladas	12
Ilustración 1. 2. Mapa conceptual del proyecto.....	12
Ilustración 1. 3. Símbolo de instalación solar fotovoltaica	14
Ilustración 1. 4. Símbolo de piscina.....	14
Ilustración 1. 5. Símbolo de riego.....	14
Ilustración 1. 6. WiFi LoRa 32 (V2). Heltec	15
Ilustración 1. 7. Representación de un servidor LAMP corriendo en Raspbian	15
Ilustración 1. 8. Icono de Android Studio.....	16
Ilustración 1. 9. Logo de SEMTECH.....	18
Ilustración 1. 10. Parámetros mensaje LoRa.....	19
Ilustración 1. 11. Programa de cálculo para LoRa SX1276	20
Ilustración 1. 12. Maqueta de simulación.....	21
Ilustración 1. 13. Conexión pin actuadores maqueta	21
Ilustración 1. 14. Conexión pin sensores maqueta	21
Ilustración 1. 15. Software simulador de inversor	22
Ilustración 1. 16. Logo ThingSpeak.....	30
Ilustración 1. 17. Logo SQL.....	31
Ilustración 1. 18. Interfaz de usuario	32
Ilustración 1. 19. WiFi LoRa 32 (V2) Heltec	34
Ilustración 1. 20. Servidor LAMP Raspberry Pi.....	35
Ilustración 1. 21. Logo Android	37
Ilustración 1. 22. Contactor.....	38
Ilustración 1. 23. Módulo relé	39
Ilustración 1. 24. Características bomba del pozo	40
Ilustración 1. 25. Relé térmico 7 a 10 A. LRD14 Schneider	40
Ilustración 1. 26. Características bomba de depuradora	41
Ilustración 1. 27. Relé térmico 5,5 a 8 A. LRD12 Schneider	41
Ilustración 1. 28. Características bomba de presión	42
Ilustración 1. 29. Relé térmico 4 a 6 A. LRD12 Schneider	42
Ilustración 1. 30. Electroválvula	43
Ilustración 1. 31. Fuente de alimentación 9V	43
Ilustración 1. 32. Conexión comunicación inversor	44
Ilustración 1. 33. Modulo TTL-RS232	44
Ilustración 1. 34. Sensor de nivel	45
Ilustración 1. 35. Diagrama de flujo programa principal finca	53
Ilustración 1. 36. Diagrama de flujo interrupción zona finca	54
Ilustración 1. 37. Interrupción LoRa zona finca.....	54
Ilustración 1. 38. Máquina de estados riego depósito.....	55
Ilustración 1. 39. Diagrama de flujo programa principal zona vivienda.....	57
Ilustración 1. 40. Interrupción timer zona vivienda	58
Ilustración 1. 41. Interrupción LoRa zona vivienda	58
Ilustración 1. 42. Raspberry Pi Model 3B+	59
Ilustración 1. 43. Logo Debian.....	60
Ilustración 1. 44. Logo DuckDNS	62

Ilustración 1. 45. Ejemplo añadir dominio DuckDNS	62
Ilustración 1. 46. Página de inicio PHPMyAdmin	64
Ilustración 1. 47. Página servidor PHPMyAdmin.....	64
Ilustración 1. 48. Aplicación Android. Solar	73
Ilustración 1. 49. Aplicación Android. Riego	74
Ilustración 1. 50. Aplicación Android. Depuradora	76
Ilustración 1. 51. Aplicación Android. Otros	78

ÍNDICE DE TABLAS

Tabla 1. 1. Ventajas y desventajas red móvil	29
Tabla 1. 2. Ventajas y desventajas LoRa.....	29
Tabla 1. 3. Trama de 12 bytes	47
Tabla 1. 4. Bytes 0 y 1. Potencia aparente	47
Tabla 1. 5. Bytes 2 y 3. Potencia activa	48
Tabla 1. 6. Byte 4. Porcentaje de batería	48
Tabla 1. 7. Byte 5. temperatura del inversor	48
Tabla 1. 8. Bytes 6 y 7. Potencia producida.....	49
Tabla 1. 9. Bytes 8 y 9. Entradas digitales	49
Tabla 1. 10. Byte 10. Salidas digitales	49
Tabla 1. 11. Byte 11. Variables de estado	50
Tabla 1. 12. Estados y transiciones	55
Tabla 1. 13. Variables máquina de estados.....	55
Tabla 1. 14. Tabla Estado.....	66
Tabla 1. 15. Tabla Inversor	66
Tabla 1. 16. Tabla TareasCíclicas	66
Tabla 1. 17. Tabla TareasCols	67
Tabla 1. 18. Tabla TCEnviada.....	67

1.1. Objeto

El objeto del proyecto es dotar de una aplicación domótica, aprovechando las nuevas tecnologías de IoT, a una instalación en una finca de recreo particular.

Recientemente, se instaló un sistema de generación de energía solar con acumuladores en la finca, la cual no dispone de acceso a la red eléctrica, lo que motivó la idea de este proyecto.



Ilustración 1. 1. Placas solares instaladas

Se pretende crear una interfaz de usuario para monitorizar la instalación fotovoltaica existente, así como controlar de forma remota y automática el sistema de riego y la depuradora de la piscina, además de otros elementos.

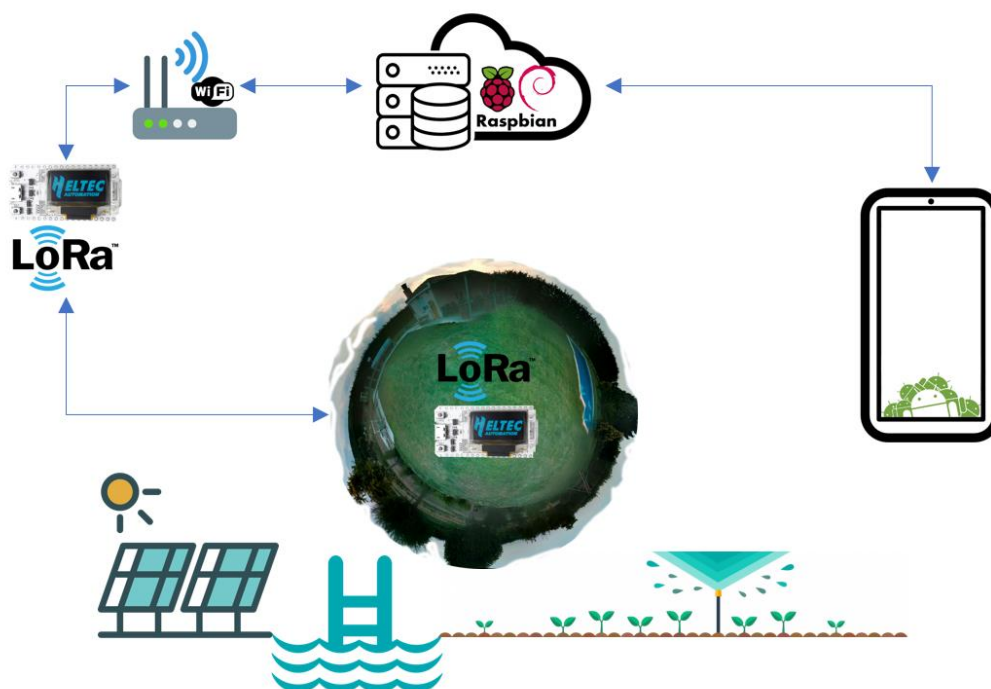


Ilustración 1. 2. Mapa conceptual del proyecto

Con la realización del proyecto se pretende obtener los siguientes beneficios:

- Aumentar la eficiencia de la instalación solar, pudiendo gestionar los excedentes y no siendo necesaria la presencia física para accionar los diferentes sistemas.
- Aumentar la vida útil de las baterías disponibles, ya que al efectuar los procesos de máxima demanda energética a las horas de máxima producción se realizan menos ciclos de descarga.
- Permitir que la piscina depure más tiempo del necesario, al no requerir de presencia física para accionarla, consiguiendo un mejor mantenimiento.
- Automatizar el riego de las distintas zonas que lo requieren.
- Mantener el nivel del depósito siempre con agua, para disponer de la misma en la casilla de la finca.

Se busca, además, realizar el proyecto de la forma más económica posible, ajustando las prestaciones del sistema a los requerimientos de la instalación sin que incremente excesivamente el precio de ésta.

1.2. Alcance

El proyecto pretende desarrollar los siguientes apartados:

- Elegir el hardware necesario para llevar a cabo los siguientes apartados:
 - Monitorización de la producción de la instalación fotovoltaica y el nivel de carga de los acumuladores.

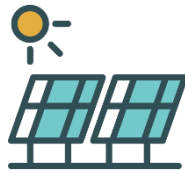


Ilustración 1. 3. Símbolo de instalación solar fotovoltaica

Para ello será necesario comunicarse vía puerto serie con el inversor que controla la instalación fotovoltaica.

- Permitir depurar de manera remota y/o automática la piscina



Ilustración 1. 4. Símbolo de piscina

Para poder realizarlo habrá que instalar un relé térmico de seguridad que pueda comprobar el microcontrolador para avisar de la avería, así como un sistema de accionamiento de la bomba de la depuradora a partir de la señal del microcontrolador

- Programar el riego para accionarlo de forma remota y/o automática



Ilustración 1. 5. Símbolo de riego

Al igual que para la piscina habrá que instalar un sistema parecido para la bomba de presión del pozo existente en la finca. Además, habrá que instalar unas electroválvulas que permitan el paso del agua hacia los aspersores de la zona correspondiente.

- Permitir añadir otros controles a automatizar, como pueden ser:
 - Nivel de un depósito de agua
 - Bomba de presión de agua
- Elección del hardware necesario para la comunicación del sistema con un servidor en internet.

Se planteará la posibilidad de utilizar una conexión directa a internet a través de una red móvil, o la utilización de una comunicación de redes de área amplia ya baja potencia, en inglés, Low Power Wide Area Network (LPWAN), hasta otro punto con acceso a internet mediante una red WiFi.

La tecnología LPWAN que elijamos deberá poder comunicarse desde la finca hasta una vivienda situada en Calahorra, a 2km en línea recta y con visibilidad directa.

- Programación del software necesario para implementar tanto el control y monitorización, como la comunicación de los datos en el sistema.

Estos últimos dos puntos se resolverán con la utilización de una tarjeta de la marca Heltec, que cuenta con un chip integrado que le dota de conectividad inalámbrica WiFi. A su vez, dispone de comunicaciones con la especificación LoRa, una de las tecnologías disponibles del tipo LPWAN, y un microcontrolador ESP32 de bajo coste. La elección se justificará en los apartados de requisitos de diseño y análisis de soluciones.

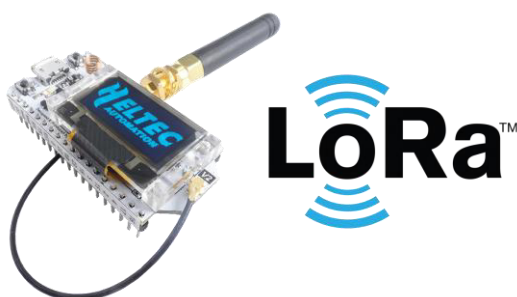


Ilustración 1. 6. WiFi LoRa 32 (V2). Heltec

- Instalación, configuración y programación del servidor alojado en una Raspberry Pi

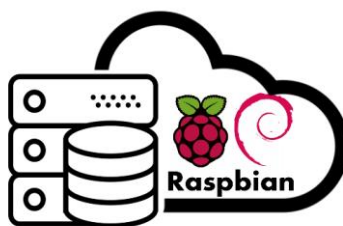


Ilustración 1. 7. Representación de un servidor LAMP corriendo en Raspbian

Se plantea la utilización de un servidor propio alojado en una Raspberry Pi frente a otras alternativas de empresas de terceros, tanto gratuitas como de pago, y se justificará la elección de utilizar un servidor propio.

- Programación de la interfaz de usuario en una aplicación para móvil

En concreto, se realizará una aplicación para Android utilizando el entorno de desarrollo gratuito oficial que proporciona Google.



Ilustración 1. 8. Icono de Android Studio

1.3. Antecedentes

La finca donde se desea realizar el proyecto no dispone de acceso a internet, pero tampoco a la red eléctrica, por lo que recientemente se realizó una instalación solar fotovoltaica para cubrir las necesidades energéticas.

Antes de contar con la energía solar, ya se disponía en la finca de los siguientes elementos:

- Grupo electrógeno diésel de 6kW de potencia
- Depuradora para la piscina con una bomba de 1,4 kW
- Bomba para extraer agua de un pozo de 1,6 kW
- Bomba de presión de agua para abastecer la casilla desde un depósito de 800 W

La instalación solar cuenta con:

- 2500W de placas solares
- Inversor de 5kW
- Acumuladores a 48V formados por 24 baterías de 2V con una capacidad de 360 Ah cada una, haciendo un total de 17,28kWh de capacidad.

Las características permiten mover perfectamente las bombas de agua disponibles, siendo la más grande de menos de 2kW.

Tras esta instalación, surge la idea de realizar este proyecto para poder gestionar los excedentes de la producción, así como mejorar la eficiencia de su uso y el mantenimiento de la finca.

Por lo tanto, al existir ya la instalación eléctrica, para la realización del proyecto solo serán necesarios unos pequeños añadidos para encender y apagar los actuadores con las señales del microcontrolador. También se añadirán algunos sensores para tener controlado el sistema y darle más seguridad.

Los elementos más significativos de la instalación anterior estarán incluidos en el apartado de otras referencias (1.4.5).

1.4. Normas y referencias

1.4.1. Disposiciones legales y normas aplicadas

En cuanto a la normativa y disposiciones legales de este proyecto, principalmente hay que tener en cuenta la legislación vigente correspondientes a las bandas ISM (industriales, científicas y médicas) en Europa.

En concreto, la banda de 868MHz, que es la utilizada para la comunicación LoRa, ya que en Europa ha sido asignada para las operaciones sin licencia.

Las limitaciones legales de esta banda se pueden resumir en dos puntos:

- No se debe emitir con una potencia superior al 25mW
- Debe respetarse una política de acceso, de modo que no se puede emitir más del 1% del tiempo.

La primera limitación no supone un problema, ya que la tarjeta seleccionada cumple con la distancia requerida en el diseño y no supera esta potencia.

Sin embargo, a la hora de realizar el diseño del protocolo de comunicación, habrá que tener en cuenta la segunda norma. Para ello se utilizará un programa de cálculo que proporciona Semtech y que se explica en el siguiente apartado.

De este modo se asegura que cada nodo no emite de media más del 1% del tiempo. Para más información y enlaces a legislación vigente consultar en el pliego de condiciones.

1.4.2. Programas de cálculo

Como se menciona en el apartado anterior, hay que tener en cuenta el tiempo del mensaje en el aire. Para esta finalidad, Semtech, empresa que posee la patente de LoRa, proporciona un software para calcularlo en función de la configuración que hagas del módulo.

La tarjeta elegida tiene un módulo SX1276, para el cual debemos utilizar la herramienta “LoRa Modem Calculator Tool”.



Ilustración 1. 9. Logo de SEMTECH

Con esta herramienta ajustamos los parámetros para que la comunicación sea lo más fiable posible, y cumpla nuestros requisitos de envío de mensajes.

Como se explicará en su apartado correspondiente al diseño del protocolo de comunicaciones, necesitamos poder enviar mensajes cada minuto.

Es decir, el módulo encargado de la monitorización y el control del sistema en la finca enviará sus datos cada minuto al módulo situado en la vivienda, para que éste los almacene en el servidor.

Por otro lado, el módulo situado en la vivienda enviará los datos cada vez que tenga una orden pendiente en la cola de tareas del servidor. Dicha cola está implementada en una de las tablas de base de datos utilizada en la aplicación.

Por lo tanto, si realizamos los cálculos para enviar una media de un mensaje por minuto, para cumplir la regla del 1% podremos enviar:

$$\frac{1msg}{min} * \frac{60min}{h} * \frac{24h}{dia} = 1440 \frac{msg}{dia}$$

Esto supone que el mensaje tendrá que ocupar la banda un máximo de 0,6s (600ms). En el anexo de cálculos se puede observar la configuración elegida y el tiempo en el aire obtenido:

- [2.2. Cálculos](#)

Esto no supone una limitación, ya que las órdenes enviadas desde la vivienda nunca van a acercarse a esa cifra.

En un uso normal del sistema sería difícil superar, en incluso acercarse, a los 20 mensajes por día, por lo que, aunque no dejemos el minuto de separación entre mensajes, nunca llegaremos al límite de ocupación de la banda del 1%.

Los parámetros que influyen en la duración del mensaje son:

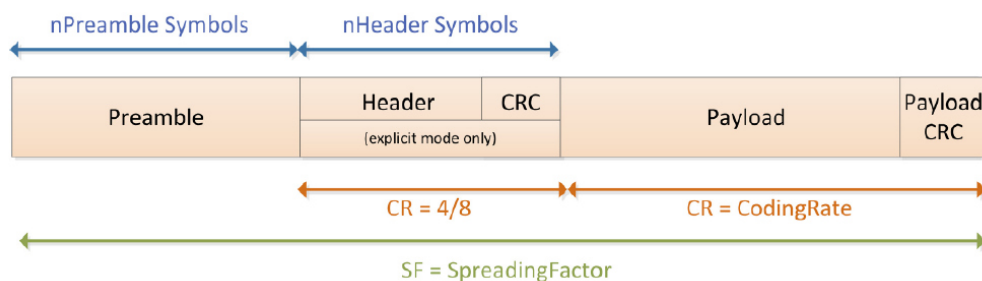


Ilustración 1. 10. Parámetros mensaje LoRa

Ajustando los parámetros como se muestra en la siguiente captura, podemos conseguir:

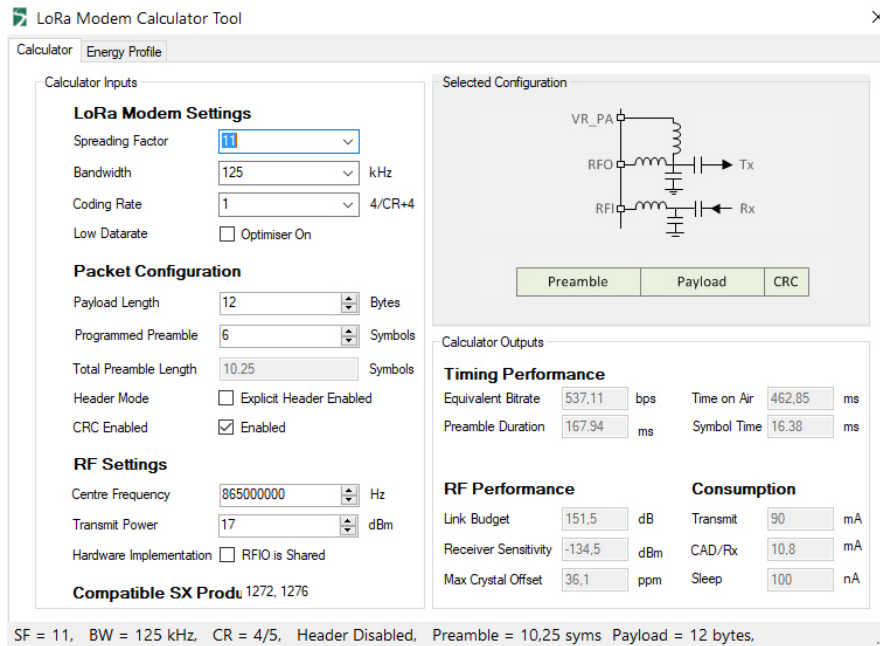


Ilustración 1. 11. Programa de cálculo para LoRa SX1276

Con esta configuración obtendríamos un tiempo en el aire del mensaje de 462,85ms, con lo que, sabiendo que este tiempo es el 1%, habrá que dejar 100 veces ese valor entre mensajes. De modo que habrá que dejar como mínimo 46,285s entre mensajes.

De este modo cumplimos el requisito de un minuto, con un margen de casi 15 segundos.

Para más información sobre la configuración de LoRa, consultar el siguiente manual vinculado en anexos y bibliografía:

- [2.2. Cálculos](#)
- [2.3. Estudio de tecnologías LPWAN](#)
- [Bibliografía 1.4.4.1. Información sobre LoRa](#)

1.4.3. Plan de gestión de la calidad aplicado durante la redacción del Proyecto

Se ha desarrollado un modelo de simulación mediante una maqueta y una aplicación para PC en LabView, con el fin de asegurar la calidad de los resultados, haciendo las pruebas pertinentes de funcionamiento.

Este modelo consiste en una maqueta, que, mediante indicadores e interruptores, simulará los actuadores y sensores del sistema (que son todos digitales).

Además, el software para PC simulará la comunicación vía puerto serie con el inversor instalado, pudiendo hacer pruebas de funcionamiento de la totalidad del sistema.

La maqueta montada para la simulación es la siguiente:

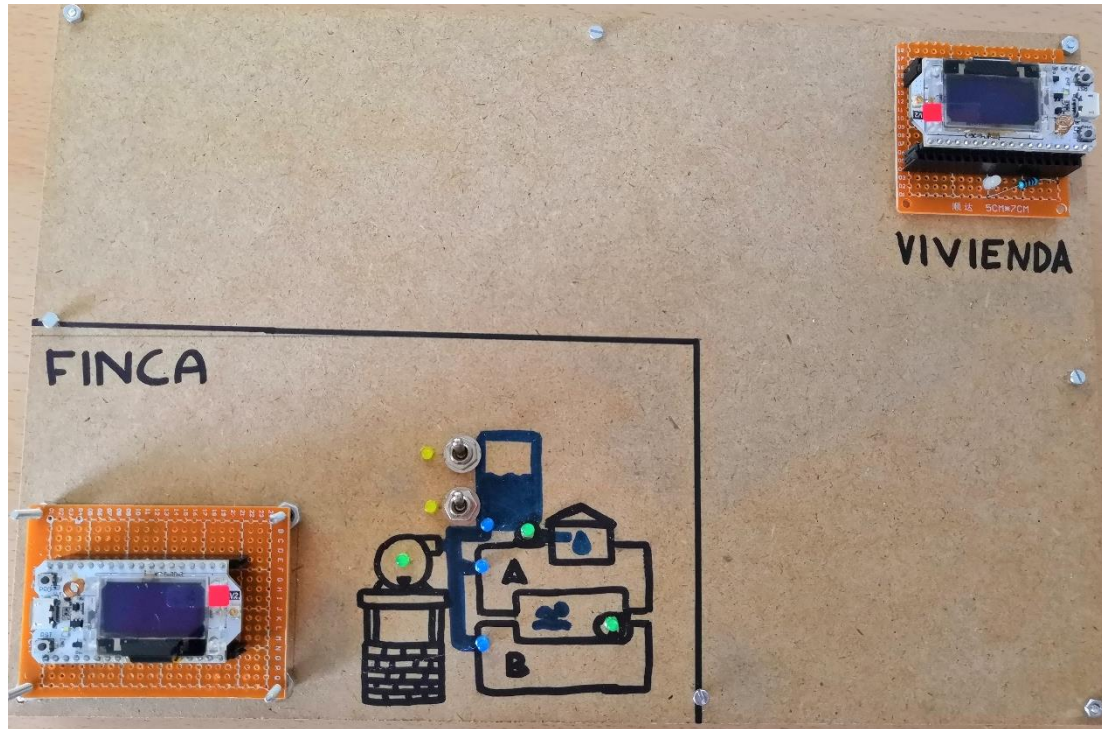


Ilustración 1. 12. Maqueta de simulación

Se pueden observar en verde las electrobombas, en azul las electroválvulas y en amarillo los sensores de nivel del depósito.

Los actuadores están conectados de la siguiente forma para la representación, siendo el PIN un pin de salida:

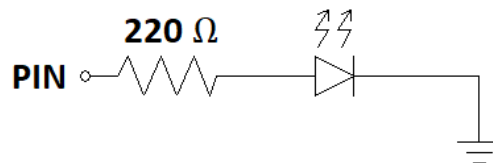


Ilustración 1. 13. Conexión pin actuadores maqueta

Mientras que los sensores utilizan la palanca para dar tensión o no al pin, y el LED es el indicador. En este caso el PIN es un pin de entrada:

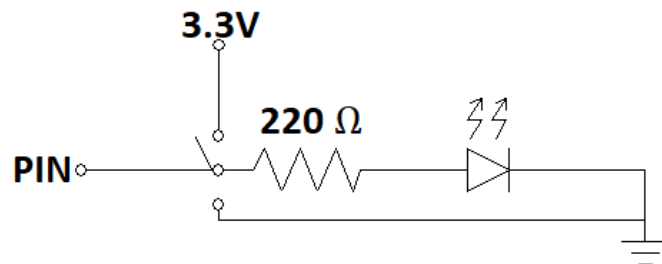


Ilustración 1. 14. Conexión pin sensores maqueta

Cada LED se corresponde a un actuador o sensor, estando conectados al pin que les corresponde siguiendo el esquema del plano 1.

- [Plano 001. Esquema WiFi LoRa 32 \(V2\).](#)

El software de simulación tiene la siguiente interfaz de usuario:

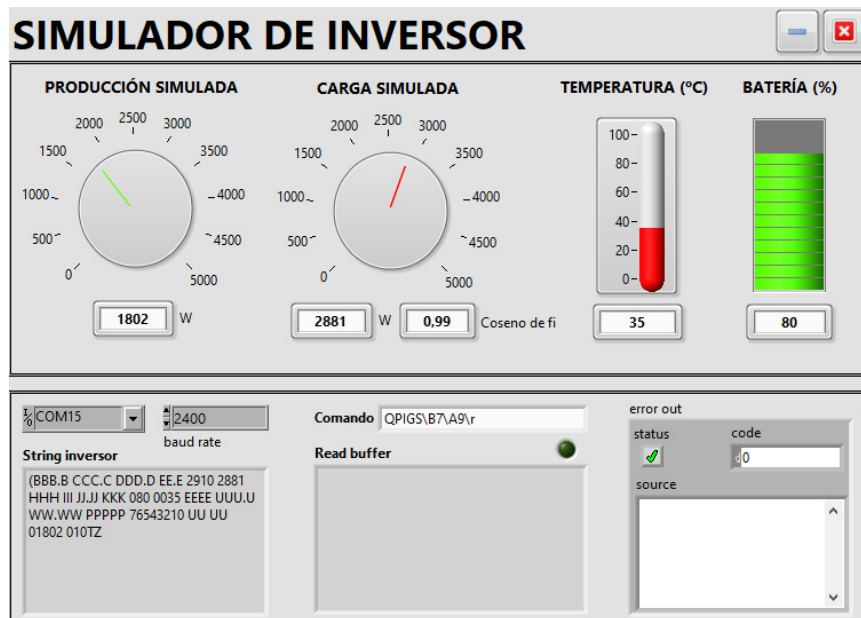


Ilustración 1. 15. Software simulador de inverter

En él, se pueden ajustar los parámetros que nos interesa medir de la trama del inverter, y también calcula la trama que generaría el verdadero equipo, mostrándola en pantalla para que podamos verla.

En la trama solo se incluyen los parámetros que podemos modificar, que son los que mediremos en realidad. Sin embargo, el resto de los parámetros que enviaría el inverter se dejan como letras, ya que no los vamos a leer.

Este software espera a recibir el comando que utiliza el inverter real, y devuelve la trama por el puerto serie. Se pueden configurar tanto el número del puerto COM, como la velocidad de transferencia (*baudrate*), utilizando por defecto el valor de 2400 baudios, la velocidad a la que comunica el real.

El código de este software se adjuntará en el anexo:

- [2.6. Modelo de simulación](#)

1.4.4. Bibliografía

1.4.4.1. Información sobre LoRa

- *What is LoRa?. Semtech Corporation.*
<https://www.semtech.com/lora/what-is-lora>
- *Low Energy Consumption Design. Semtech Corporation, 2013*
https://www.semtech.com/uploads/documents/LoraLowEnergyDesign_STD.pdf
- *LoRa Modem Designer's Guide. Semtech Corporation, 2013*
https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf
- *SX1276 Development Kit. Semtech Corporation, 2014*
https://www.semtech.com/uploads/documents/SX1276DevKit_Userguide_STD.pdf

1.4.4.2. ESP32, Arduino y librerías LoRa

- *Página web de Arduino.*
<https://www.arduino.cc/>
- *Documentación ESP32. Espressif Systems. 2019.*
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- *WiFi LoRa 32 V2 Information. Heltec*
<https://heltec.org/project/wifi-lora-32/>
- *Heltec ESP32 Based Dev-boards Develop Environment.*
https://docs.heltec.cn/#/en/user_manual/how_to_install_esp32_Arduino
- *GitHub. Librería WiFi Kit Series.*
https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series

1.4.4.3. Información sobre el inversor presente en la instalación

- *Inversor Axpert. Inversor Híbrido MKS II 48v 5000w Mppt 80 500Vdc.*
<https://www.rebacas.com/home/640-inversor-hibrido-axpert-mks-ii-48v-5000w-mppt-80-500v.html>

1.4.4.4. *Raspberry Pi y servidor*

- *Hardware. RaspberryPi.org.*
<https://www.raspberrypi.org/documentation/hardware/>
- *Linux. RaspberryPi.org.*
<https://www.raspberrypi.org/documentation/linux/>
- *Raspbian. RaspberryPi.org.*
<https://www.raspberrypi.org/documentation/raspbian/>
- *Prometec.*
<https://www.prometec.net/indice-raspberry-pi/>
- *Documentación PHP.*
<https://www.php.net/manual/es/intro-what-is.php>
- *Documentación MySQL.*
<https://dev.mysql.com/doc/>
- *Documentación PHPMyAdmin.*
<https://www.phpmyadmin.net/docs/>
- *Documentación Apache.*
<https://httpd.apache.org/docs/>

1.4.4.5. *Android Studio*

- *Android Developers. Guía de usuario.*
<https://developer.android.com/studio/intro>
- *MPAndroidChart. Librería de gráficos.*
<https://github.com/PhilJay/MPAndroidChart>
- *GSON. Librería para archivos json de Google.*
<https://github.com/google/gson>
- *Librería retrofit. HTTP Client para Android.*
<https://square.github.io/retrofit/>

1.4.5. Otras referencias

- *Datasheet MAX3232.*
<http://descargas.cetronic.es/MAX3232.pdf>
- *Datasheet módulo relé.*
<https://docs-emea.rs-online.com/webdocs/13b5/0900766b813b5d9e.pdf>
- *Datasheet contactor.*
http://www.farnell.com/datasheets/2334949.pdf?_ga=2.185289184.83651909.1566143192-1527176688.1566143192&_gac=1.186796890.1566143192.CjwKCAjw7uPqBRBIEiwAYDsr13Uswlu4yu-ItRBbCWIVZtybcUYeRY1s1kDMBmepzaiGqe8h994dRBoCfQ8QAvD_BwE
- *Datasheet relé térmico 7 a 10 A.*
http://www.farnell.com/datasheets/2044018.pdf?_ga=2.186360544.83651909.1566143192-1527176688.1566143192&_gac=1.90785128.1566143192.CjwKCAjw7uPqBRBIEiwAYDs r13Uswlu4yu-ItRBbCWIVZtybcUYeRY1s1kDMBmepzaiGqe8h994dRBoCfQ8QAvD_BwE
- *Datasheet relé térmico 4 a 6 A.*
http://www.farnell.com/datasheets/2044023.pdf?_ga=2.186360544.83651909.1566143192-1527176688.1566143192&_gac=1.90785128.1566143192.CjwKCAjw7uPqBRBIEiwAYDs r13Uswlu4yu-ItRBbCWIVZtybcUYeRY1s1kDMBmepzaiGqe8h994dRBoCfQ8QAvD_BwE
- *Datasheet sensor de nivel.*
https://www.digikey.es/product-detail/es/cynergy-3/RSF86H100R/725-1132-ND/2835331?utm_adgroup=Float%2C+Level+Sensors&mkwid=sDBqslid2p&pcrid=315558758615&pkw=&pmt=&pdv=c&productid=2835331∓slid=&gclid=CjwKCAjw-7LrBRB6EiwAhh1yX99hgZ-OnoC9NxQefTvRI_jqHEgE32wtaFU6KNux2FpQ4DGpwLLhbXoClksQAvD_BwE
- *Electroválvulas Hunter.*
<https://www.hunterindustries.com/es/product/valvulas/pgv>

1.5. Definiciones y abreviaturas

Finca: Se trata de una finca rústica con una casilla, una piscina y una huerta. Es una instalación privada no industrial que se utiliza para ocio. También puede referirse a la finca como: huerto o terreno.

Casilla: Caseta pequeña con una cocina y un cuarto de baño ubicada en la finca sobre la que se realiza el proyecto.

LPWAN: Low Power Wide Area Network, en español, redes de área amplia y baja potencia. Es un protocolo de transmisión inalámbrica de datos, básico hoy en día para aplicaciones IoT.

LoRa: acrónimo de LongRange. Es una técnica de modulación de espectro extendido derivada de la tecnología chirp spread spectrum (CSS). Esta tecnología es una de las disponibles para las comunicaciones LPWAN.

Baudrate: tasa de baudios, Es el número de unidades de señal por segundo. Habitualmente se puede equiparar a los bits por segundo, pero no siempre se mantiene la relación, ya que depende de cuantos niveles tenga la señal enviada.

LAMP: acrónimo que describe un sistema de infraestructura de internet, que se apoya en las siguientes herramientas de software libre: Linux, Apache, MySQL/MariaDB y Perl/PHP/Python.

WAMP: alternativa al sistema LAMP que sustituye Linux por Windows.

DNS: Domain Name System, en español, sistemas de nombre de dominio. Sistema jerárquico de nomenclatura descentralizado, utilizado en redes IP.

DDNS: DNS dinámico. Es un servicio que permite mantener una actualización en tiempo real de los nombres de dominio.

Iv4: Protocolo de internet versión 4. Las direcciones en este formato son 4 números de 1 byte, por lo que puede haber hasta 4,294,967,296 direcciones. Son las direcciones que estamos acostumbrados a utilizar en redes locales (ej. 192.168.0.1).

IPv6: Protocolo de internet versión 6. En este caso se aumenta el número de direcciones a 8 números de 16 bits, permitiendo hasta 340,282,366,920,938,463,463,374,607,431,768,211,456 direcciones. Este inmenso número de direcciones está pensado para admitir el crecimiento de internet en futuras generaciones.

JSON: acrónimo de JavaScript Object Notation, en español, Notación de objeto JavaScript. Se trata de un formato de texto para el intercambio de datos en forma de subconjunto de objetos de JavaScript.

API: Application Programming Interface, en español, interfaz de programación de aplicaciones. Es un conjunto de funciones o subrutinas que ofrece una biblioteca la cual puede ser utilizada por otro software a modo de capa de abstracción.

1.6. Requisitos de diseño

1.6.1. El cliente

Esta instalación está pensada para una finca particular con cinco propietarios. Tiene un uso principalmente de ocio, y cuenta además con una pequeña huerta.

El proyecto podría extrapolarse a instalaciones agrónomas pequeñas donde haya que controlar determinados actuadores semejantes a los que tiene un sistema de riego, y monitorizar una serie de sensores.

1.6.2. Emplazamiento y entorno socioeconómico y ambiental

En este caso la instalación va a complementar una ya existente, otorgándole la posibilidad de monitorizar y controlar remotamente el sistema, que actualmente es manual, por lo que no va a existir un impacto medioambiental adicional al ya existente.

La finca está situada en la provincia de La Rioja, entre Calahorra y Murillo de Calahorra a unos 2 kilómetros en línea recta de la ciudad, y no dispone de acceso a la línea eléctrica ni a internet, salvo con la utilización de la red móvil.

1.6.3. Estudios encaminados a la definición de la solución adoptada

Al plantearse el proyecto se abren un amplio abanico de posibilidades para llegar a una solución apropiada.

1.6.3.1. *Estudio de alternativas para la comunicación con Internet*

La primera decisión importante era la forma de acceder a internet para intercambiar los datos de monitorización y control.

No se dispone de conexión en la finca, por lo básicamente existen dos alternativas:

- Conectarse a internet mediante una red móvil
- Comunicar los datos hasta una vivienda próxima a la finca donde ya se dispone de conexión.

La primera solución las

Comunicación con internet mediante red móvil	
VENTAJAS	DESVENTAJA
Reducir tiempos de programación	Incrementa el coste de mantenimiento del sistema, ya que implica una tarifa fija asociada a la operadora de red móvil
Evitar una comunicación adicional de los datos	
Simplificar el diseño del sistema	

Tabla 1. 1. Ventajas y desventajas red móvil

En cuanto a la segunda, sería necesario utilizar una comunicación Low Power Wide Area Network (LPWAN), ya que existe una distancia a la vivienda mencionada de unos 2km en línea recta, contando con visibilidad directa.

Se estudiaron diferentes alternativas para esta comunicación, y se optó por una comunicación punto a punto de LoRa. El estudio de la comparativa entre este tipo de tecnologías se adjunta en anexos.

- [2.3. Estudio de tecnologías LPWAN](#)

Comunicación de los datos con LPWAN LoRa hasta una vivienda con acceso a internet	
VENTAJAS	DESVENTAJAS
Reduce el coste de mantenimiento de la red, ya que solo es necesario adquirir los 2 módulos de LoRa con comunicación punto a punto	Necesario diseñar una comunicación
La tecnología LoRa está pensada para este tipo de proyectos de IoT	Sistema algo más complejo
Comunicaciones de largo alcance (hasta 20km)	Limita la velocidad de transmisión de datos
Bajo consumo de energía	
Soporta muy bien las interferencias	
Utiliza bandas de frecuencia libres de licencia (868MHz en Europa)	

Tabla 1. 2. Ventajas y desventajas LoRa

Para más información sobre LoRa consultar el anexo de la comparativa realizada entre tecnologías LPWAN, donde se informa sobre este tipo de tecnologías. También se podrá encontrar más información en los enlaces de la bibliografía.

1.6.3.2. Estudio de las alternativas del almacenamiento de datos en la nube

Para almacenar los datos de control y de monitorización en internet se puede optar por varias soluciones.

La primera alternativa planteada consiste en utilizar un servicio de almacenamiento de datos para IoT en versiones gratuitas.

En las versiones gratuitas nos encontramos limitaciones en distintos aspectos, dependiendo de la empresa que ofrezca el servicio.

La opción más interesante que ha sido analizada era ThingSpeak, una plataforma de almacenamiento de datos pensada para IoT, con una gran integración con Matlab.

La plataforma permite graficar y publicar los datos almacenados y es interesante para esta clase de usos.

La principal limitación de esta clase de servicios gratuitos es la rápida caducidad de los mismos, lo que implica que con cierta periodicidad deberemos hacer una copia de los datos y volver a crear las tablas (canales en el caso de ThingSpeak), incrementando considerablemente el mantenimiento.



Ilustración 1. 16. Logo ThingSpeak

También suelen tener limitación en el número de actualizaciones de los datos, lo que dificulta el control del sistema, mientras que por las características del sistema podría servir para la monitorización.

En ThingSpeak existe una versión de pago, partiendo de los 75€ anuales la versión de uso doméstico más básica, mientras que la configuración más básica para uso comercial parte de los 600 anuales (precios de agosto de 2019).

Otra alternativa era montar un servidor de bases de datos con software libre y desarrollar una base de datos SQL. De esta forma se eliminan las limitaciones de uso de las aplicaciones desarrolladas por empresas de terceros, mencionadas en apartados anteriores.



Ilustración 1. 17. Logo SQL

Para tener un servidor existen a grandes rasgos dos alternativas

- Utilizar un servicio de hosting alquilando los servidores a una empresa.
- Montar un servidor propio ubicado en la vivienda mencionada en el proyecto

En ambos casos se va a requerir de bastante más tiempo de programación que en la primera, siendo, siendo la solución más laboriosa la de montar un servidor propio.

Alquilar servidores de terceros puede conseguirse por un precio desde los 5-10€ mensuales hasta miles de euros en función de las prestaciones, aunque bastaría con los servicios más básicos para la aplicación en cuestión.

Para la opción del servidor propio tenemos a su vez la posibilidad de montar un servidor con unas prestaciones relativamente altas, o bien utilizar un ordenador de bajas prestaciones, pero suficiente para la aplicación en cuestión.

Las limitaciones del servidor propio montado en la vivienda sería principalmente la estabilidad de la conexión a internet, dependiendo de la instalación en la vivienda. Sin embargo, actualmente, con la fibra óptica, esta limitación solo sería notable en caso de corte de suministro de la red eléctrica.

También habría que seleccionar, en caso de optar por esta solución, el sistema operativo del servidor, así como el tipo de servidor que se monte.

Siendo la opción más interesante implementar un servidor que integra una infraestructura mínima de comunicaciones Internet con las siguientes herramientas

Este servidor estaría formado por.

- Un sistema operativo Linux
- Un servidor web Apache
- Un servidor de bases de datos MySQL o MariaDB
- Un lenguaje de programación de propósito general PHP

LAMP es un paquete que integra la infraestructura anterior y está basado en software libre.

Existen otras alternativas como WAMP (que sustituye a Linux por Windows), Microsoft SQL Server y otros.

1.6.3.3. Alternativas de la interfaz de usuario

Para la interfaz de usuario disponemos de varias posibilidades, que podrían ser implementadas en su conjunto, pero de las cuales seleccionaremos una, por tiempo y economía.

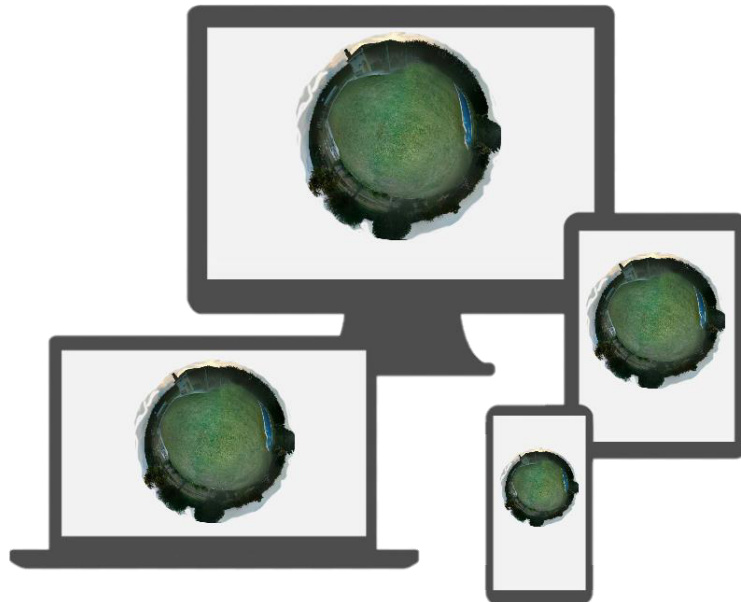


Ilustración 1.18. Interfaz de usuario

Interfaz de usuario en local

Esta interfaz no es muy lógica implementarla como solución única, ya que perdería el sentido el sistema de comunicaciones planteado anteriormente. Sería una opción interesante como complemento.

Esta solución consistiría en un panel de control local (con botonera y varios indicadores o una pantalla) situado en la finca, y que permitiera la monitorización y el control desde un punto cómodo en la casilla de la finca.

Interfaz de escritorio

Consistiría en una aplicación para PC desde la que se realizaría el control y la monitorización. Esta solución no es la más práctica, ya que requiere de un PC para realizar las funciones, y se trata de una aplicación de domótica aplicada a un entorno rural, por lo que lo más cómodo sería utilizar un Smartphone para el control.

Interfaz en aplicación web

Sería una aplicación muy interesante de cara a compatibilidad con todas las plataformas con acceso a internet, y disponiendo de un servidor propio sería viable. La principal desventaja es el rendimiento y la estética en comparación con una aplicación nativa del sistema.

Interfaz en aplicación móvil

Esta solución es prácticamente igual de cómoda que la aplicación web, aunque menos versátil en compatibilidad con sistemas operativos. Puesto que los usuarios de la finca utilizan en su totalidad dispositivos Android, sería interesante realizar una aplicación para este sistema operativo.

Tanto la aplicación web como la móvil, son las dos claras candidatas para ser seleccionadas como únicas interfaces de usuario con el sistema (aparte de las conexiones físicas, que seguirán existiendo en modo manual).

Puesto que, en su mayoría, por no decir únicamente, se va a acceder desde el móvil, y los tiempos de programación son similares a priori para ambas aplicaciones, la opción que parece más interesante es la aplicación Android, de cara a la estética y la eficiencia en el funcionamiento.

1.7. Análisis de soluciones

1.7.1. Elección de las comunicaciones y el hardware necesario

Teniendo en cuenta las soluciones posibles, se ha optado por una tecnología de comunicaciones LPWAN que es más económica a largo plazo. Se trata de una red de comunicación LoRa punto a punto, con un nodo en la vivienda y otro en la finca.

El nodo colocado en la vivienda será el encargado de conectarse con el servidor y servir de punto de intercambio entre éste y el otro nodo.

En el otro extremo de la red, el módulo ubicado en la finca por su parte será el encargado del control de los sensores y actuadores necesarios para cumplir los requisitos del diseño, y se comunicará con la vivienda para recibir los datos del servidor y guardar los datos tomados.

A pesar de las limitaciones de utilizar la comunicación Low Power Wide Area Network (LPWAN) con LoRa, principalmente en cuanto a velocidad de transmisión, esta es suficiente para las características buscadas, por lo que compensa utilizarlo por el ahorro a largo plazo.

Por lo tanto, se necesita elegir un hardware que nos permita controlar y medir señales digitales, y que a su vez sea compatible con LoRa y con capacidad para acceder a internet en la vivienda (bien sea mediante una antena WiFi o por ethernet).

Existen varias alternativas en el mercado, pero para el desarrollo del proyecto se adquirieron dos tarjetas de Heltec, en concreto las WiFi LoRa 32 (V2).



Ilustración 1. 19. WiFi LoRa 32 (V2) Heltec

Esta tarjeta cuenta con:

- Microcontrolador ESP32 a 80MHz, que permite realizar de forma holgada el control requerido.
- WiFi integrado, lo que nos permite conectarnos a internet en la vivienda sin necesidad de adquirir un módulo aparte.
- La tarjeta cuenta como es necesario, con un módulo LoRa integrado, en este caso el SX1276, de la marca Semtech (empresa con la patente de LoRa). El circuito integrado es compatible con la banda de uso libre de licencias en Europa, la banda de 868MHz.

- Como extra, esta tarjeta cuenta con una pantalla OLED de 0,96 pulgadas, que en un principio no va a utilizarse, pero permitirá incluir una pequeña interfaz de usuario en local en un futuro.

La principal ventaja de esta tarjeta es la compatibilidad con el entorno de desarrollo de Arduino, permitiendo utilizar las librerías ya existentes para ESP32, así como unas propias para esta tarjeta en concreto, facilitando la tarea de programación.

Además, el precio de la tarjeta es inferior a los 20 dólares en la página oficial de Heltec, y en torno a los 20€ en Amazon, por lo que no se trata de un precio elevado incluyendo todo el hardware necesario en una misma tarjeta.

Existen otras alternativas sin pantalla OLED, algo más económicas, que también podrían utilizarse para el proyecto.

La decisión de utilizar las tarjetas mencionadas vino pensando en utilizar la pantalla para futuras mejoras, así como para facilitar las tareas de desarrollo del software, pudiendo utilizarlas en las pruebas.

Los pines digitales de esta tarjeta funcionan a 3,3V, por lo que habrá que elegir el hardware apropiado para adaptar las señales digitales disponibles y poder activar las electroválvulas que y los contactores de las bombas de presión.

1.7.2. Elección del servidor

Para el almacenamiento e intercambio de datos a través de internet se ha decidido crear un servidor LAMP (Linux, Apache, MySQL, PHP) en una Raspberry Pi Model 3 B+, abierto a internet mediante un servicio de DDNS.

La principal ventaja de este tipo de servidor es la utilización de software libre, lo que permite tener todas las características de forma gratuita.

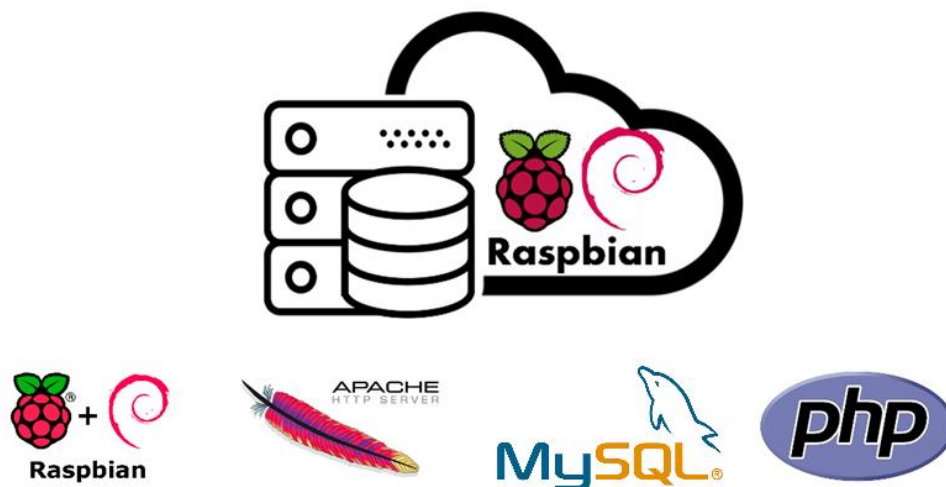


Ilustración 1. 20. Servidor LAMP Raspberry Pi

La elección concreta será:

- Raspbian como distribución de Linux para Raspberry Pi
- Apache como servidor web
- MySQL como gestor de bases de datos
- PHP como lenguaje de programación

Cuenta con Linux como sistema operativo, Apache como servidor web, MySQL como gestor de bases de datos, y PHP como lenguaje de programación.

El motivo de esta elección ha sido las posibilidades que ofrece tener un servidor propio en cuanto a acceso, versatilidad y no estar sujeto a posible caducidad de los servicios, así como el coste de mantenimiento.

Por ello, finalmente se ha optado por la solución más económica que permita cumplir con los requisitos mínimos del proyecto planteado.

Las desventajas de esta elección son:

- La estabilidad de la conexión al servidor, que está sujeta a los cortes de electricidad e internet de la vivienda donde se va a alojar.
- Las limitaciones de procesamiento y almacenamiento de una Raspberry Pi.

En un primer momento no se va a tomar una solución para las posibles caídas del servidor que inutilizarían la aplicación de interfaz de usuario.

Una posible solución sería colocar un servidor redundante en otro emplazamiento con conexión a internet, haciendo muy difícil que se caigan ambos a la vez, o tener un servicio gratuito de emergencia trabajando en segundo plano. Este problema también se podría solucionar añadiendo al servidor un pequeño SAI comercial, o un sistema de baterías autónomo diseñado para tal fin.

Sin embargo, al tratarse de una aplicación no industrial, no sería demasiado problemática una caída de los servidores, y se incrementaría el costo de la instalación, por lo que se ha estimado que no es necesario cubrir este posible fallo.

De cara a las limitaciones de procesamiento, al tratarse de un servidor únicamente utilizado para esta aplicación, y con un número muy limitado de usuarios, tratándose de consultas sencillas al servidor, que principalmente será intercambio y almacenamiento de datos en SQL, no debería verse limitado en potencia a pesar de sus características sencillas.

1.7.3. Elección de la interfaz de usuario

Para liberar carga del servidor se decidió que la interfaz de usuario no estaría ubicada directamente en él (desarrollando una aplicación web), y se optó por la solución más práctica, desarrollar una aplicación Android para poder controlar el sistema desde un smartphone.



Ilustración 1. 21. Logo Android

Se eligió Android como sistema operativo sobre la que correr la aplicación debido a varios factores:

- La mayoría de los usuarios disponen de dispositivos con este sistema operativo.
- El entorno de desarrollo para Android es accesible desde PC, mientras que iOS solo puede programarse desde MAC.
- Se prefiere un sistema operativo móvil sobre uno de escritorio para esta clase de tareas, donde interesa más la movilidad que una pantalla grande.
- Una aplicación web habría sido interesante por dar mayor cobertura de dispositivos, pero requeriría más carga del servidor y al tratarse de uno de bajas prestaciones la experiencia de usuario mejora utilizando una aplicación propia para Android.
- No poder utilizar la aplicación desde un PC no es problemático, ya que la información es accesible a través de la página de PHPMYAdmin instalada en el servidor, por lo que en caso de necesitar acceder a ella desde otra plataforma siempre tenemos esta posibilidad. Se utilizará principalmente para mantenimiento del sistema y mejoras a añadir en el futuro.

1.8. Resultados finales

1.8.1. Modificación de la instalación eléctrica para el control y la monitorización del sistema

Para la realización del proyecto es necesaria una pequeña modificación en la instalación eléctrica. Será necesario añadir los siguientes elementos:

- Para el control y monitorización de las bombas de las bombas
 - Relés de 3,3V de continua que permite al microcontrolador dar señal de mando a los contactores trifásicos de los motores.
 - Contactores para los motores que se colocarán paralelos a la instalación manual existente (ya que esta instalación es complementaria, no sustitutiva)
 - Relés térmicos para las bombas
- Para el control y monitorización del agua (riego y depósito)
 - Electroválvulas en paralelo a los grifos utilizados actualmente para abrir los circuitos de agua necesarios para el riego y el llenado del depósito.
 - Relés u otros elementos para adaptar los 3,3V del microcontrolador a la señal de mando de las electroválvulas.
- Para la monitorización del inversor
 - Adaptación de los pines utilizados para la comunicación serie de TTL a RS232

Todos los elementos elegidos a continuación tienen las características técnicas definidas en el siguiente apartado del pliego de condiciones:

- [4.2.2. Especificaciones de los elementos constitutivos del proyecto.](#)

MOTORES DE LAS BOMBAS DE PRESIÓN

La modificación no es grande, por lo que se requiere de poco tiempo de desarrollo y de instalación.

En primer lugar, vamos a seleccionar un contactor adecuado para cada una de las bombas.

Las bombas que tenemos que controlar tienen una potencia de 800W la más pequeña y 1,1kw la más grande, por lo que hay que buscar un contactor con una capacidad de carga superior.

Se ha elegido un contactor de la marca Telemecanique que puede mover motores de hasta 2,2kw en tensiones de 230V.



Ilustración 1. 22. Contactor

Se adjuntará en referencias un enlace a su ficha técnica. Se va a utilizar por simplificar el mismo contactor para las tres bombas. [1.4.5. Otras referencias.](#)

Una vez seleccionado el contactor, vamos a elegir los relés que nos den la señal de mando para los contactores seleccionados, en este caso 230V.

Como solo va a dar señal de mando no hace falta que sea capaz de soportar grandes cargas, por lo que la mayoría nos valdrán.

En este caso necesitamos acondicionar la señal de 3,3V de continua a los 230V de alterna de la red para dar mando los 3 contactores elegidos. El siguiente relé cumple los requisitos mencionados:

Para el relé vamos a emplear 3 módulos de relé con tensión de alimentación de 5V, pero cuya señal de entrada puede ser de 3,3V. La alimentación a 5V podríamos hacerla desde el pin de 5V que tiene la tarjeta de Heltec utilizada, pero por seguridad se va a conectar a una fuente de 5V externa.

Además, utilizando una fuente externa se libera consumo de la tarjeta, que está pensada para dar señales de control, mientras que la bobina del relé puede consumir bastante en relación con lo que aporta la placa.

La principal ventaja sin embargo de utilizar una alimentación externa es aprovechar que se trata de un relé con optoacoplador, por lo que aislaríamos por completo a nivel eléctrico la placa del relé.

El módulo elegido es el siguiente:

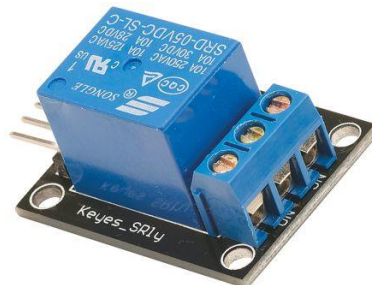


Ilustración 1. 23. Módulo relé

Si miramos su ficha técnica vemos las siguientes características:

- Controla cargas hasta 10A
- Provee de aislamiento entre el controlador y la carga
- Tensión de alimentación 5VDC
- Tensión de entrada 3.3-5VDC
- Su corriente máxima a la salida es 10A a 250VAC o 30VDC

Para más información: [1.4.5. Otras referencias.](#)

Para alimentar a los relés se utilizará el pin de 5V de la tarjeta LoRa, que estará alimentada por micro USB con una fuente que admite hasta 2A de carga. Si miramos las especificaciones el pin de 5V admitirá una carga igual a la entrada que tenga por el conector micro USB (restándole el consumo de la tarjeta.)

Por último, vamos a incluir un relé térmico de protección para las bombas, por lo que deberemos seleccionar el adecuado para cada una.

La bomba más potente tiene una corriente nominal de 7 A, tiene la siguiente tabla de características:

Q min/max	2.5/8	m ³ /h	Outlet	1 ¼"
H max/min	48/15	m	n	2900 r/min
P	1.6	kw/ 2.15	HP	C 25 μF/ 450 V
	1~220 V/	50 Hz	7 A	W.T 18.5 kg
IP68	ICL F	S1	S/N	18010009

Ilustración 1. 24. Características bomba del pozo

Por lo tanto, buscaremos un relé de sobrecarga apropiado. Para este caso utilizaremos el siguiente de la marca Schneider, que ofrece protección de 7 a 10 A regulables.



Ilustración 1. 25. Relé térmico 7 a 10 A. LRD14 Schneider

Las otras dos bombas tienen un consumo más bajo, por lo que habrá que buscarles la protección adecuada.

La bomba de la piscina tiene las siguientes características:

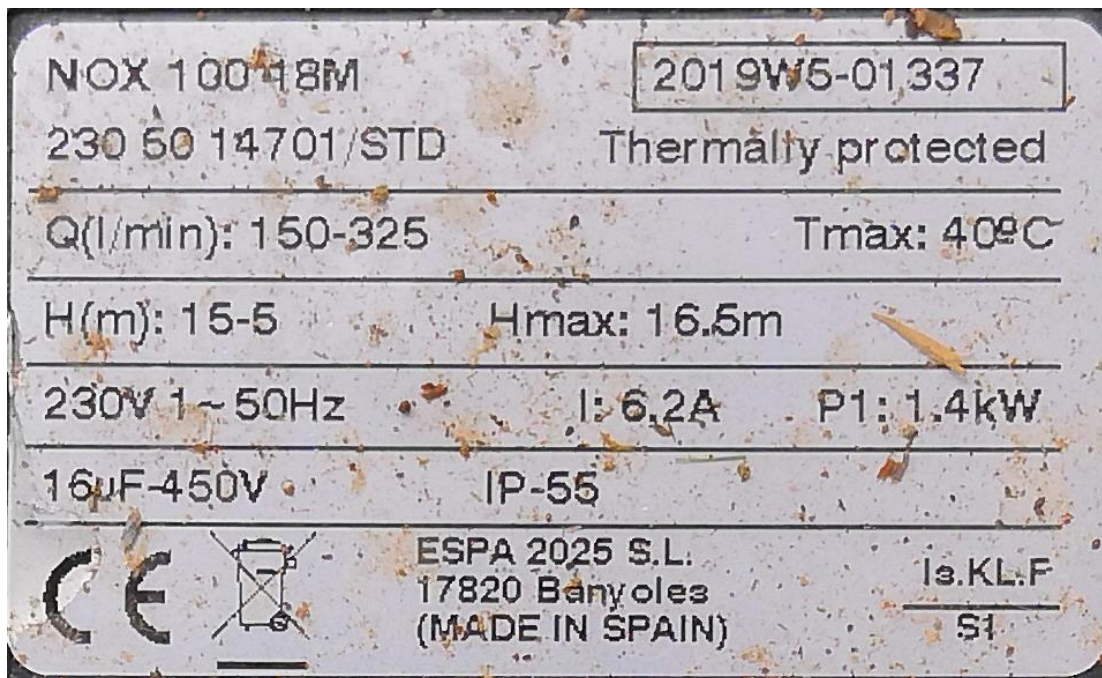


Ilustración 1. 26. Características bomba de depuradora

Por lo tanto, hay que buscar una protección de 6,2 A. Se va a utilizar un relé térmico de la misma gama que el anterior, el LRD12, que ofrece protección entre 5,5 y 8 A:



Ilustración 1. 27. Relé térmico 5,5 a 8 A. LRD12 Schneider

Por último, la bomba de presión de agua de la casilla tiene las siguientes características:

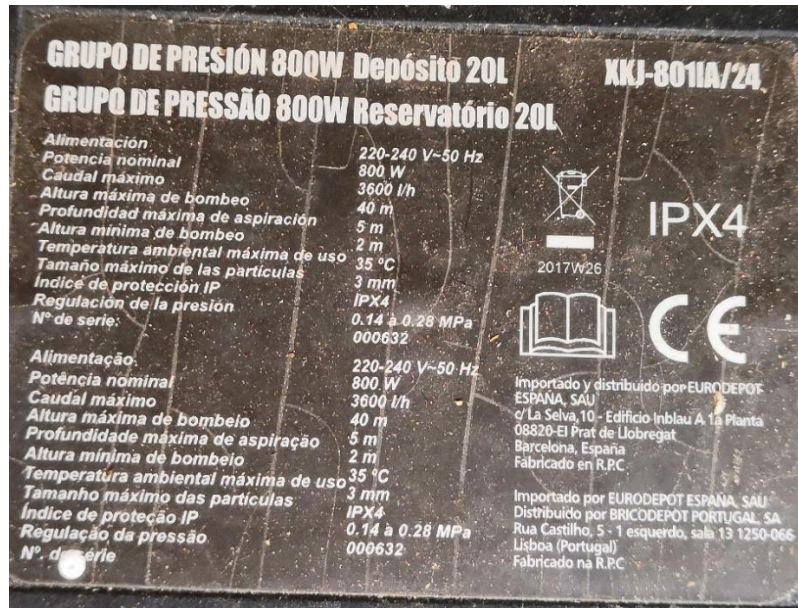


Ilustración 1. 28. Características bomba de presión

Para este caso la protección debe ser de unos 4 A, para ello se elige el siguiente relé. Se trata de un modelo de la misma gama y marca que el anterior, pero con la protección apropiada para esta bomba.



Ilustración 1. 29. Relé térmico 4 a 6 A. LRD12 Schneider

Con estos puntos ya tendríamos la modificación necesaria para el control y monitorización de las bombas.

Para ver las conexiones necesarias consultar los planos de la instalación eléctrica.

- [Plano 002. Esquema de control](#)
- [Plano 003. Esquema de potencia](#)

AGUA (RIEGO Y DEPÓSITO)

Para esta modificación vamos a buscar unas electroválvulas apropiadas para controlar el paso del agua.

El caudal aproximado que debe ser capaz de manejar cada una es de unos 7000l/h, que es el que aporta la bomba de presión a la que irán conectadas.

Para cumplir estos requisitos se ha seleccionado la siguiente electroválvula:



Ilustración 1. 30. Electroválvula

Funciona a 9V por lo que deberemos utilizar 3 relés más para activar los solenoides. Los relés seleccionados funcionaban tanto en alterna como en continua, por lo que se utilizarán los mismos, necesitando un total de 6.

Esta electroválvula tiene las siguientes características:

- Tensión de funcionamiento 9V
- Caudal mínimo 0,7l/min
- Caudal máximo 150l/min = 9000l/h

En los planos se incluye la instalación de estas 3 electroválvulas en el circuito hidráulico de la instalación, así como sus conexiones eléctricas.

Para cubrir la tensión de funcionamiento de las electroválvulas se va a utilizar una fuente de alimentación de 9V como la siguiente:



Ilustración 1. 31. Fuente de alimentación 9V

Una fuente de 9V y 1A que podrá proporcionar la energía suficiente para activar las electroválvulas conectándola a los relés de control.

La información sobre los componentes seleccionados se encuentra de forma más exhaustiva en el pliego de condiciones. [4.2.2. Especificaciones de los elementos constitutivos del proyecto.](#)

MONITORIZACIÓN DEL INVERSOR

El inversor cuenta con un puerto de comunicación serie con interfaz RS232 pero con conector rj45. Con él, ya hay incluido un cable conversor rj45 a RS232 db9.

Como disponemos del cable que venía incluido y de cables rj45 estándar, se va a optar por colocar las dos adaptaciones para los pines TTL, ya que los conectores necesarios son baratos, y así podemos utilizar ambos cables.

En el manual de las comunicaciones del inversor se especifica las patillas utilizadas en la conexión del puerto serie con el que nos vamos a conectar al inversor, y descrita en la siguiente imagen:

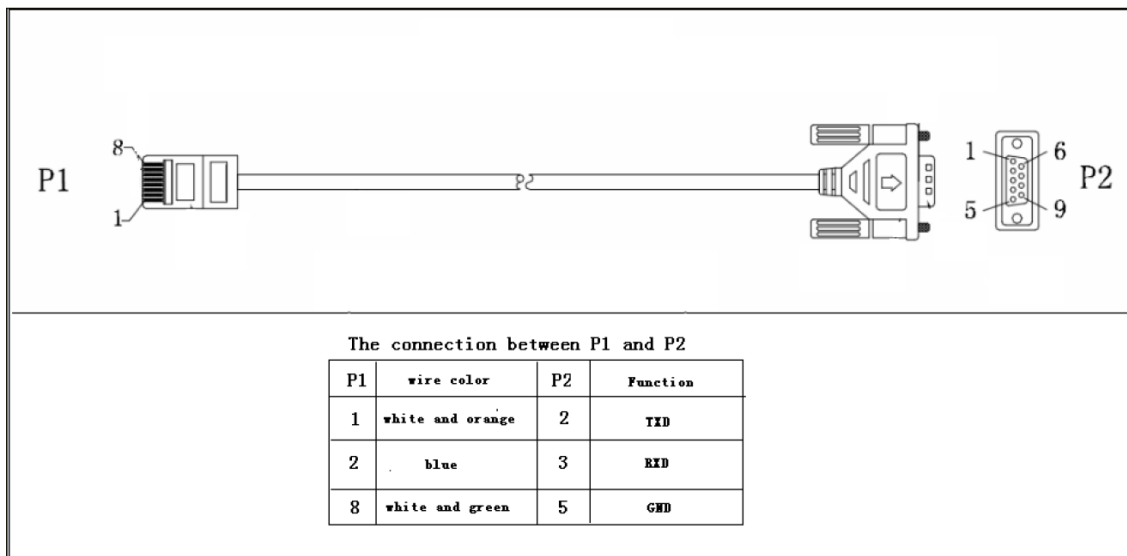


Ilustración 1. 32. Conexión comunicación inversor

En primer lugar, es necesario adaptar la señal TTL a RS232. Para ello la solución elegida consta de una versión del chip MAX3232 y se trata del siguiente módulo:

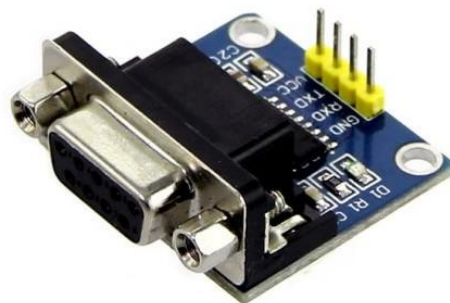


Ilustración 1. 33. Módulo TTL-RS232

Se trata de un módulo ya preparado que permite disponer de una interfaz rs232 con conector DB9 y compatible con el microcontrolador elegido del ESP32.

El módulo viene ya con la interfaz RS232, por lo que podremos conectar los pines de la pcb directamente a los correspondientes en un conector hembra de rj45 y de este modo ya contaríamos con los 2 puertos.

El inconveniente de este módulo es su conector integrado para rs232, que es hembra al igual que el cable incluido, por lo que será necesario un adaptador macho-macho db9 null modem (ya que debe cruzar las líneas) para poder conectar por este puerto.

Por lo tanto, para adaptar la conexión serie utilizaremos:

- Módulo MAX3232 con db9 hembra integrado
- Adaptador rs232 macho-macho null modem de 9 pines
- Conector hembra rj45 (opcional)

CONTROL DE NIVEL DEL DEPÓSITO

Por último, los sensores de nivel utilizados son de tipo flotador, que actúan como un interruptor que se cierra al flotar en el agua. Se trata del siguiente sensor:



Ilustración 1. 34. Sensor de nivel

El esquema eléctrico viene adjuntado en los planos, donde se detalla exactamente que pines será necesario conectar al módulo.

- [Plano 001. Esquema WiFi LoRa 32 \(V2\).](#)

1.8.2. Comunicación LoRa punto a punto

Como se ha mencionado en apartados anteriores, la comunicación de los datos hasta el punto de acceso a internet va a hacerse mediante LoRa.

Se puede consultar el estudio sobre la comunicación LPWAN en el anexo sobre ello, y las referencias de la bibliografía para obtener una idea más completa de la tecnología LoRa:

- Anexo. [2.3. Estudio de tecnologías LPWAN](#)
- Bibliografía. [1.4.4.1. Información sobre LoRa](#)

A modo de resumen:

LoRa es un sistema de modulación de radiofrecuencia patentado por Semtech. Tiene una gran capacidad para realizar conexiones seguras distancias grandes. Se suele aplicar a redes de Internet de las Cosas, donde se necesitan dispositivos de bajo consumo.

Se trata de una tecnología LPWAN (Low Power Wide Area Network), es decir, para redes de largo alcance y baja potencia (bajo consumo).

Esta tecnología es ideal para esta clase de sistemas, en los que hay mala o ninguna conexión a internet y se desea crear una red privada de sensores y actuadores.

En este caso se desarrolla una conexión punto a punto propia, sin utilizar operadores de red, ya que se busca minimizar el coste de mantenimiento.

1.8.2.1. Configuración del módulo

Con la librería para ESP32 es muy sencillo colocar los valores elegidos para la configuración de LoRa, ya que consta de funciones incluidas en la librería para tal cometido.

Información sobre la librería mencionada:

- Bibliografía. [1.4.4.2 ESP32, Arduino y librerías LoRa](#)

La configuración elegida para cumplir con los requisitos planteados es la siguiente:

- **Spreading factor o tiempo de esparcimiento.** Se ha seleccionado en 11, siendo su valor máximo 12 y el mínimo 6. Cuanto mayor es este valor más tiempo dura el mensaje en el aire, haciéndolo más robusto a interferencias, pero aumentando también el tiempo necesario a dejar entre mensajes. Es un factor de escala que afecta a todo el mensaje.
- **Bandwidth.** Es el ancho de banda que ocupa el mensaje. Cuanto menor, más lenta es la comunicación, pero siguiendo el manual de diseño de Semtech es conveniente elegir los 125kHz frente a las otras dos opciones (250 y 500kHz) para factores de esparcimiento altos.
- **Coding rate.** Se trata de una implementación para corregir errores de comunicación, se puede configurar entre 1 y 4 entre, afectando a un divisor, por lo que con un valor de 1 tenemos mayor coding rate. Para el caso se ha seleccionado 1.
- **Preamble.** Se trata de un preámbulo programable previo al mensaje principal. Se utiliza principalmente en redes LoRaWAN, y su valor mínimo es 2. La configuración se ha dejado como estaba por defecto en 6, dejando un margen mayor de maniobra.
- **Payload.** Duración del mensaje. Se ha establecido en 12 bytes, que son los mensajes más largos que se van a enviar siguiendo el protocolo establecido a continuación

Los valores de *Spreading factor*, *Bandwidth* y *coding rate* seleccionados son los que tiene por defecto la librería utilizada.

Con esta configuración, como se mostraba en el apartado de programas de cálculo, el mensaje tiene un tiempo en el aire de 462,85ms, por lo que, si de media podemos enviar el 1% del tiempo, esto se traduce en un máximo de 1866 mensajes al día (teniendo en cuenta un mensaje de 12 bytes).

1.8.2.2. Protocolo desarrollado

El protocolo que se ha desarrollado para la comunicación es el siguiente:

Tenemos dos nodos que se comunican punto a punto. Por un lado, el nodo ubicado en la finca es el encargado de aportar la información sobre el estado de los sensores y actuadores del sistema.

Esta información recogida por el nodo se enviará periódicamente cada minuto al otro nodo, de modo que este tiempo queda establecido y es fijo.

El envío cada minuto se realiza para cumplir las normas sobre la ocupación de la banda de 868MHz. Más información sobre estas limitaciones y la configuración elegida para cumplirla en el apartado de normas y referencias (1.4).

En ese apartado se fija que los mensajes deberán tener como mucho 12 bytes para cumplir las comunicaciones cada minuto dejando un margen.

Por lo tanto, del lado del nodo ubicado en la finca tendremos una trama de 12 bytes, la cual estará diseñada para enviar el máximo de información en tan poco tamaño. Estos bytes contienen los datos explicados en las tablas a continuación:

- **Trama de 12 bytes**

Bytes de la trama enviada cada minuto desde la finca											
0	1	2	3	4	5	6	7	8	9	10	11
VA	WC		%Bat	Tª Inv.	WP		Digital IN		D.OUT	Estado	
El microcontrolador toma muestras del inversor 1 vez por segundo y va generando el valor medio de las muestras del último minuto, cada vez que toma 60 muestras las envía junto con el estado de sensores, actuadores y variables de estado importantes											

Tabla 1. 3. Trama de 12 bytes

- **Bytes 0 y 1**

VA - Potencia aparente consumida en VA															
Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Parte entera														Decimales	
Valor entre 0 y 16383														0 = 0	
														1 = 0,25	
														2 = 0,5	
														3 = 0,75	

Tabla 1. 4. Bytes 0 y 1. Potencia aparente

- Bytes 2 y 3

WC - Potencia activa consumida en W															
Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Parte entera														Decimales	
Valor entre 0 y 16383														0 = 0	
														1 = 0,25	
														2 = 0,5	
														3 = 0,75	

Tabla 1. 5. Bytes 2 y 3. Potencia activa

- Byte 4

% Bat - Porcentaje de las baterías							
Bits							
7	6	5	4	3	2	1	0
Parte entera							Decimales
Valor entre 0 y 127 (realmente entre 0 y 100)							0 = 0
							1 = 0,5

Tabla 1. 6. Byte 4. Porcentaje de batería

- Byte 5

Tª Inv. - Temperatura del inversor							
Bits							
7	6	5	4	3	2	1	0
Parte entera							Decimales
Valor entre 0 y 127							0 = 0
							1 = 0,5

Tabla 1. 7. Byte 5. temperatura del inversor

- Bytes 6 y 7

WC - Potencia producida en W															
Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Parte entera													Decimales		
Valor entre 0 y 16383													0 = 0		
													1 = 0,25		
													2 = 0,5		
													3 = 0,75		

Tabla 1. 8. Bytes 6 y 7. Potencia producida

- Bytes 8 y 9

Digital IN - Medida de los sensores digitales															
Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B0: Error válvula A* B1: Error válvula B* B2: Error válvula depósito* B3: Error de Inversor B4 y B5: Indefinidos* B6: Térmico bomba de presión del pozo B7: Térmico bomba de presión de depuradora B8: Térmico bomba de presión del depósito B9: Nivel depósito lleno B10: Nivel depósito vacío Resto: Indefinidos*															
*Están pensados para poder añadir más sensores de seguridad en futuras actualizaciones en función de los fallos que se produzcan durante los primeros meses de la instalación															

Tabla 1. 9. Bytes 8 y 9. Entradas digitales

- Byte 10

D. OUT - Variables digitales de salida para el control de actuadores							
Bits							
7	6	5	4	3	2	1	0
x	x	Válvula llenado depósito	Válvula riego B	Válvula riego A	Bomba presión de agua	Bomba depuradora	Bomba pozo

Tabla 1. 10. Byte 10. Salidas digitales

- **Byte 11**

ESTADO - Variables de estado importantes (estado de las configuraciones)							
Bits							
7	6	5	4	3	2	1	0
Libres		Depurando	Llenando depósito	Regando lado B	Regando lado A	Presión automática	Llenado automático del depósito

Tabla 1. 11. Byte 11. Variables de estado

En cuanto al nodo ubicado en la vivienda no se seguirá un envío periódico, si no que el ESP32 realizará una petición al servidor, que le responderá con la siguiente tarea a enviar en la cola de tareas (tabla explicada en el apartado de desarrollo del servidor. [1.8.4.2. Tablas de la base de datos](#), [Tabla 1.16. TareasCola](#))

En caso de existir una tarea pendiente, envía la orden correspondiente por LoRa. El número de órdenes que se estiman a enviar a lo largo de un día es muy difícil que se supere las 20, mientras que la normativa y la configuración elegida nos permiten mandar hasta casi 2000 al día. De este modo la orden se envía instantáneamente tras ser leída, sin incumplir el uso legal de la banda, disminuyendo el tiempo de respuesta del sistema.

El nodo en casa dispondrá de un contador de segundos que se pondrá a 0 una vez reciba la trama del nodo en la finca.

Este contador inhabilita el envío de órdenes si es superior a 55 segundos, dejando 5 segundos de margen para recibir el siguiente mensaje, momento en el que vuelve a ponerse a 0.

Si el contador supera los 180 segundos se activa el error de comunicación LoRa y se indica en el servidor.

Este protocolo sirve para evitar que se envíen datos los dos nodos a la vez y se pierdan los mensajes. Otra solución sería emplear otros dos nodos con y dos canales distintos dentro de la misma banda. De este modo también se evita pisar los mensajes, pero se duplica la instalación, por lo que como no es crítico que pueda haber un retraso de unos 5 segundos en realizar la orden, se ha decidido realizar este protocolo.

Además, en base a las distintas pruebas que se vayan haciendo una vez esté el sistema en funcionamiento, este margen de 5 segundos podrá ajustarse para tener menos posibilidades de retrasos en la ejecución.

Las órdenes posibles para enviar se muestran a continuación, todas ocupan 3 o 4 bytes, teniendo espacio de sobra para los 12 bytes sobre los que se estimó el tiempo en el aire del mensaje.

Existen dos tipos de órdenes:

- Órdenes de 4 bytes: Incluyen 3 bytes de identificación y 1 byte de valor de la orden

- Órdenes de 3 bytes. Solo envían la identificación y esto implica una acción en el sistema.

ÓRDENES DE 4 BYTES

En este tipo de órdenes el cuarto byte indica un valor concreto que acompaña a la orden. Este valor puede ser:

- Un temporizador entre 0 y 255 minutos (para controlar acciones que deben realizarse durante un tiempo)
- Un valor booleano, 0 o 1, indicando un parámetro de configuración del sistema

Las órdenes de 4 bytes programadas actualmente son:

- **RRAX**
 - RRA → Regar lado A
 - X → Duración en minutos 0-255
- **RRBX**
 - RRB → Regar lado B
 - X → Duración en minutos 0-255
- **RABX**
 - RAB → Regar ambos lados
 - X → Duración en minutos 0-255
- **DEPX**
 - DEP → Depurar
 - X → Duración en minutos 0-255
- **CFLX**
 - CFL → Configurar llenado automático
 - X:
 - 0: Manual
 - 1: Automático
- **CFPX**
 - CFp → Configurar presión automática
 - X:
 - 0: Manual
 - 1: Automático

ÓRDENES DE 3 BYTES

Funcionan solo con el identificador, y son utilizadas para llevar a cabo acciones concretas, que no dependen de otros parámetros.

Las órdenes de 3 bytes programadas actualmente son:

- LLD → Llenar depósito
- STR → Parar riego
- STD → Parar depuradora
- STL → Parar llenado
- RST → Reiniciar valores del sistema

Añadir nuevas órdenes en futuras modificaciones sería sencillo, aunque habría que tenerlo en cuenta tanto en el software de control, como en la API y la aplicación de interfaz de usuario, y sería la forma de comunicar las nuevas funcionalidades entre todas las partes del sistema.

1.8.3. Software de control para ESP32

Las dos tarjetas con el ESP32, LoRa y WiFi, ubicadas una en la finca y otra en la vivienda, deberán encargarse de lo siguiente:

- Tarjeta ubicada en la finca
 - Controlar actuadores y monitorizar sensores
 - Comunicarse con el inversor
 - Enviar los datos por LoRa
 - Recibir las órdenes por LoRa
- Tarjeta ubicada en la vivienda
 - Comprobar la cola de tareas del servidor
 - Enviar las tareas pendientes por LoRa
 - Recibir los datos de LoRa
 - Actualizar los datos en el servidor

En ambos casos en cada ciclo principal del ESP32 se actualiza la pantalla OLED con variables que muestran información sobre el estado de conexión. Para más información sobre los datos mostrados consultar el pliego de condiciones:

- [4.2.3. Condiciones y comprobación de funcionamiento](#)

1.8.3.1. Software en el módulo de la finca

En esta tarjeta Heltec se ha desarrollado un programa que consta de una serie de máquinas de estados y dos interrupciones utilizada para programar un temporizador (timer) de 1 segundo y programar el código asociado a la interrupción de recepción de paquetes de comunicación en la red LoRa.

TIMER DE 1 SEGUNDO

El código incorporado al temporizador de un segundo será el que controle la secuencia de comunicación con el inversor, tomando muestras regulares cada segundo. Además de tener un contador, que al llegar a 60, calculará la media de los 60 valores tomados, y enviará la trama de 12 bytes a la vivienda.

INTERRUPCIÓN LORA

El código asociado a la interrupción de LoRa recibirá las órdenes de la vivienda, actuando directamente sobre las máquinas de estados que controlan las salidas digitales. Estas máquinas también funcionarán teniendo en cuenta las medidas de los sensores.

DIAGRAMA DE FLUJO Y MÁQUINA DE ESTADO

Los diagramas de flujo general del programa con sus interrupciones son los siguientes:

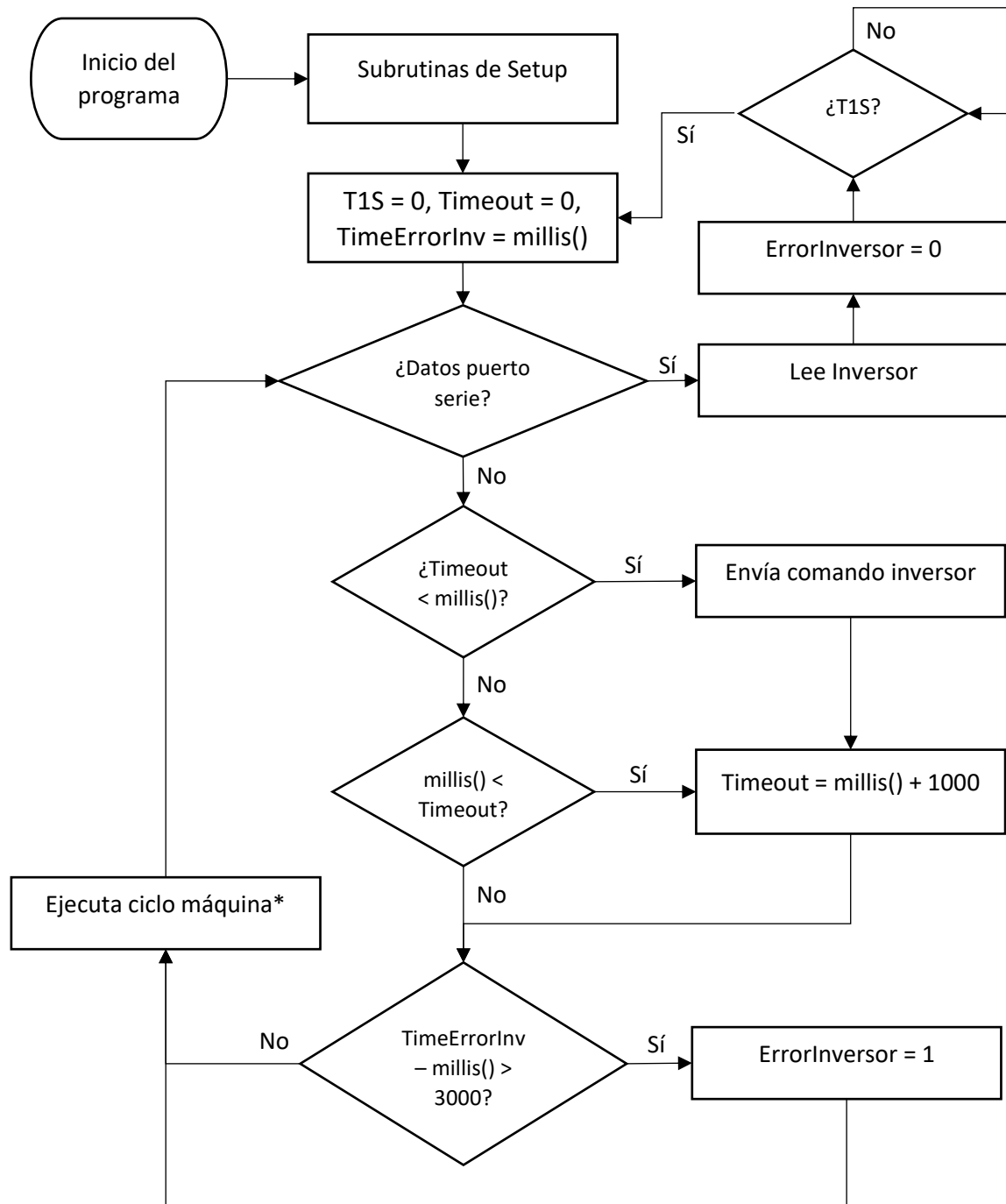


Ilustración 1. 35. Diagrama de flujo programa principal finca

*La función de ejecuta ciclo máquina lee las entradas, comprueba acciones automáticas, ejecuta un ciclo de la máquina de estados de riego y pozo, actualiza depuradora y presión, y por último escribe las salidas.

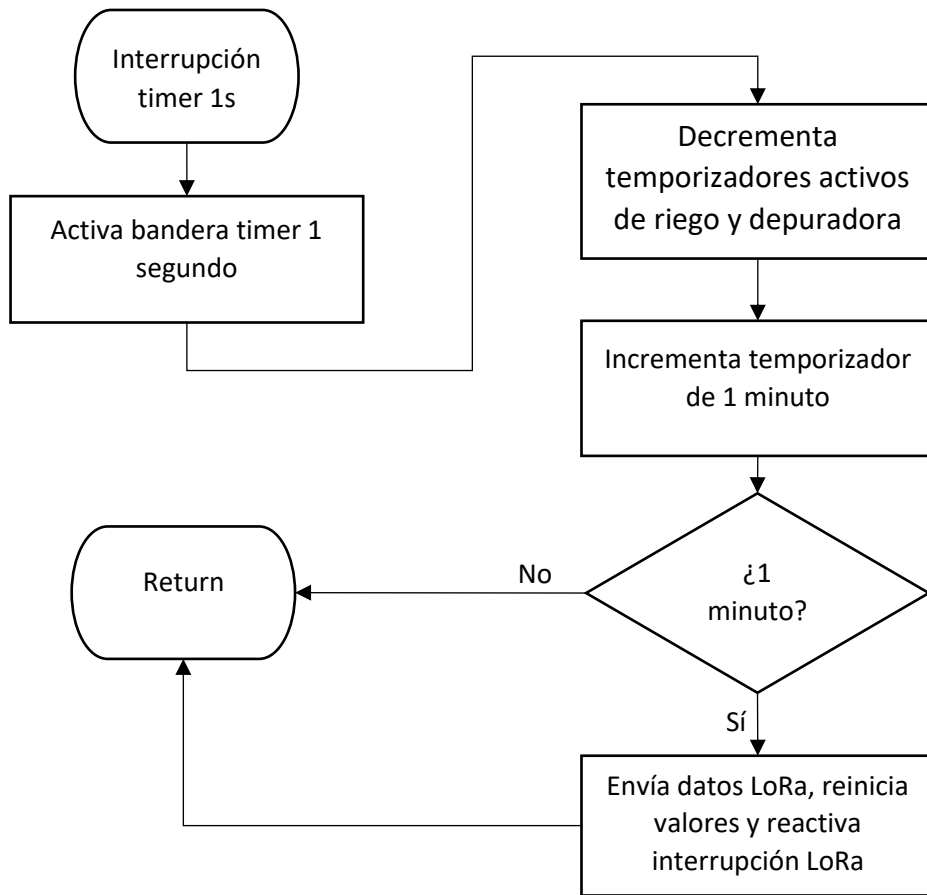


Ilustración 1. 36. Diagrama de flujo interrupción zona finca

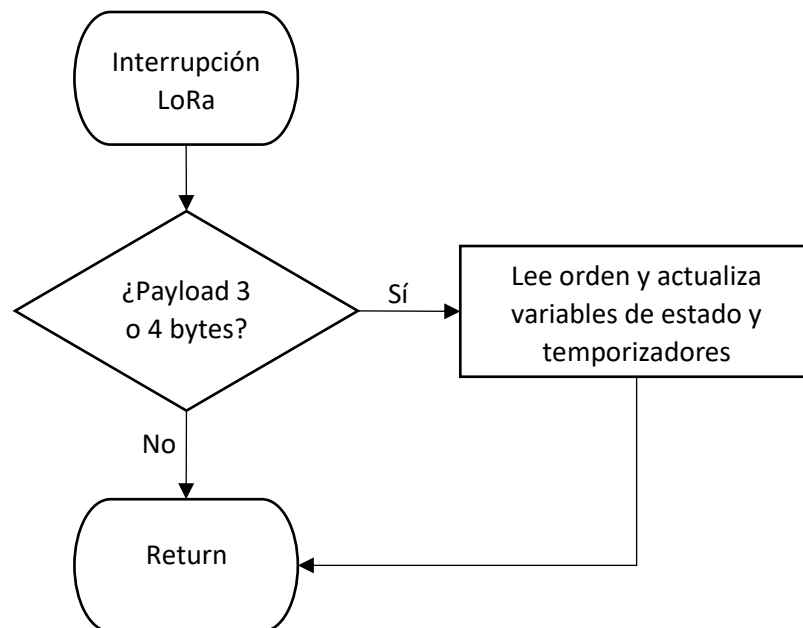


Ilustración 1. 37. Interrupción LoRa zona finca

La máquina de estados se utiliza para gestionar de forma más eficiente la bomba del pozo, encargada del riego y llenado del depósito.

Se trata de una máquina tipo Mealey, ya que las salidas dependen no solo del estado, también de las entradas.

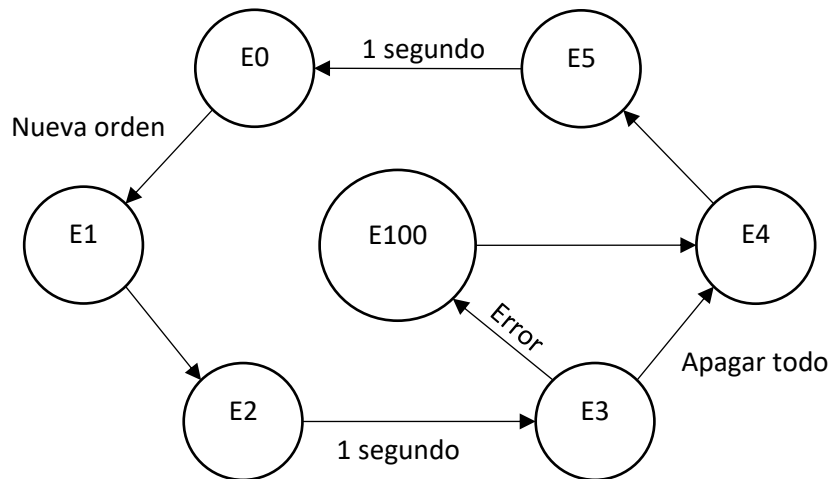


Ilustración 1. 38. Máquina de estados riego depósito

Estado	Descripción
E0	Estado de reposo, todo cerrado
E1	Pone un timeout de 1 segundo
E2	Abre válvula correspondiente
E3	Enciende bomba del pozo
E4	Pone un timeout de 1 segundo
E5	Apaga bomba pozo
E100	Error - Resetea las órdenes y temporizadores

Tabla 1. 12. Estados y transiciones

*En lugar de medir el relé térmico se va a calcular el fallo por relé térmico midiendo los consumos que otorga el inversor, y comparándolos con los que debería haber, de este modo se simplifica el esquema eléctrico.

Variable	Tipo	Descripción
TBP	Sensor	Fallo relé térmico
RA	Variable	Orden de regar A
RB	Variable	Orden de regar B
LD	Variable	Orden de llenar depósito
RegandoA	Variable	Bandera de riego A
RegandoB	Variable	Bandera de riego B
LlenandoD	Variable	Bandera de llenado depósito
EVA	Actuador	Electroválvula A
EVB	Actuador	Electroválvula B
EVD	Actuador	Electroválvula D
BP	Actuador	Bomba pozo

Tabla 1. 13. Variables máquina de estados

*También afectan indirectamente los temporizadores configurados al iniciar la orden, ya que al llegar a 0 desactivan su orden relacionada.

El código completo se adjuntará electrónicamente y estará en anexos.

- Anexo. [2.7. Código desarrollado](#)
- Anexo. [2.7.2.1. ZonaFinca](#)

1.8.3.2. Software en el punto de conexión a internet

En este nodo existirá también un timer de 1 segundo, la interrupción de recepción LoRa y el programa principal.

TIMER DE 1 SEGUNDO

El timer contará el tiempo desde la última recepción LoRa, para controlar errores en la comunicación y el protocolo explicado en el apartado anterior. Inicialmente estará configurado con un valor 1 segundo superior al establecido para el corte del envío.

INTERRUPCIÓN LORA

Además, también tendrá la interrupción LoRa para la recepción de los datos, que calculará la trama a enviar siguiendo la API desarrollada para guardar los datos (explicada en el desarrollo del servidor). A su vez, activará una bandera que indicará nuevo mensaje y pondrá el timer a 0, indicando que se pueden enviar órdenes y no hay error de comunicación.

El código completo se adjuntará electrónicamente y estará en anexos.

- Anexo. [2.7. Código desarrollado](#)
- Anexo. [2.7.2.2. ZonaVivienda](#)

DIAGRAMA DE FLUJO

Los diagramas de flujo general del programa con sus interrupciones son los siguientes:

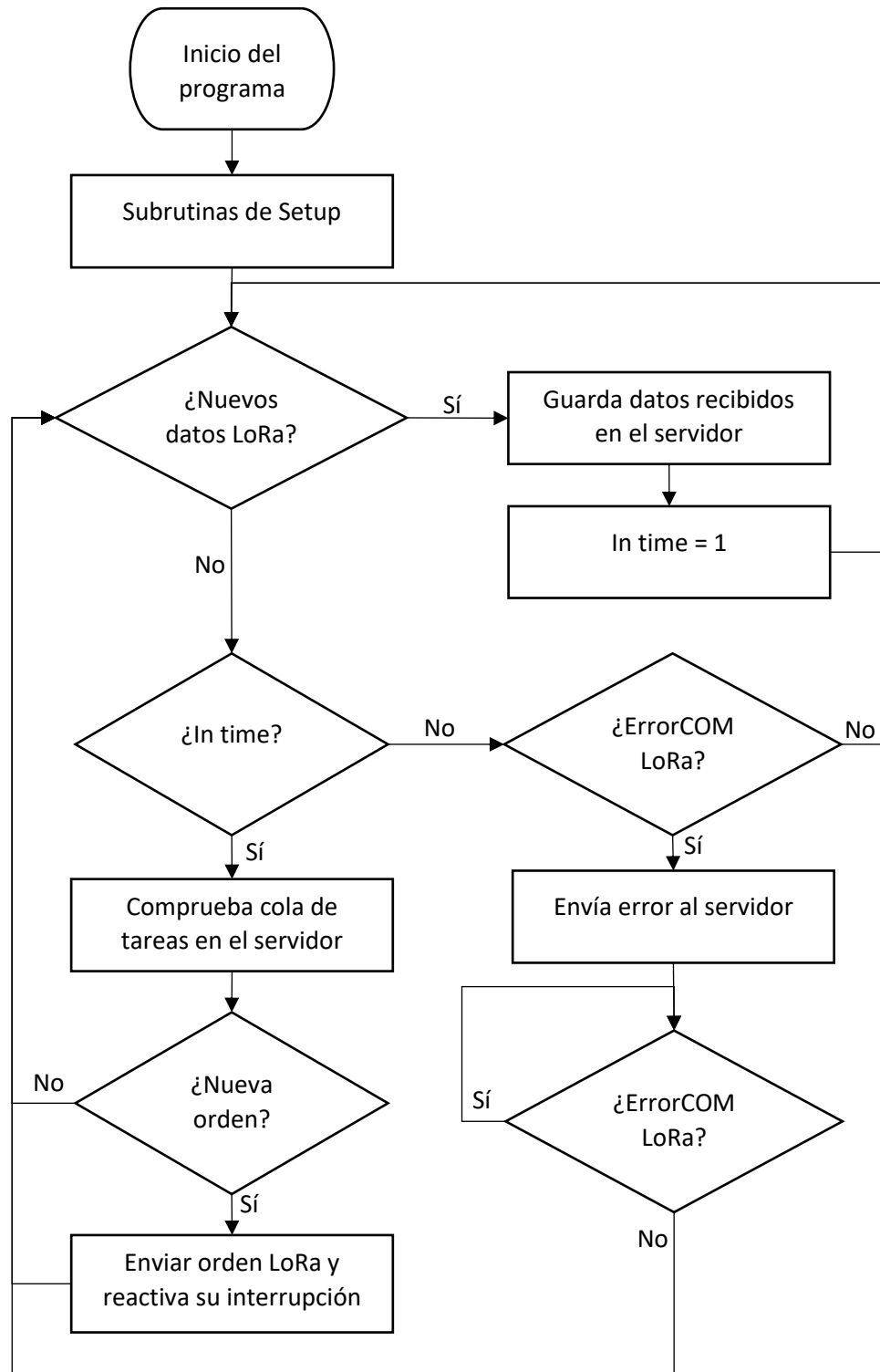


Ilustración 1. 39. Diagrama de flujo programa principal zona vivienda

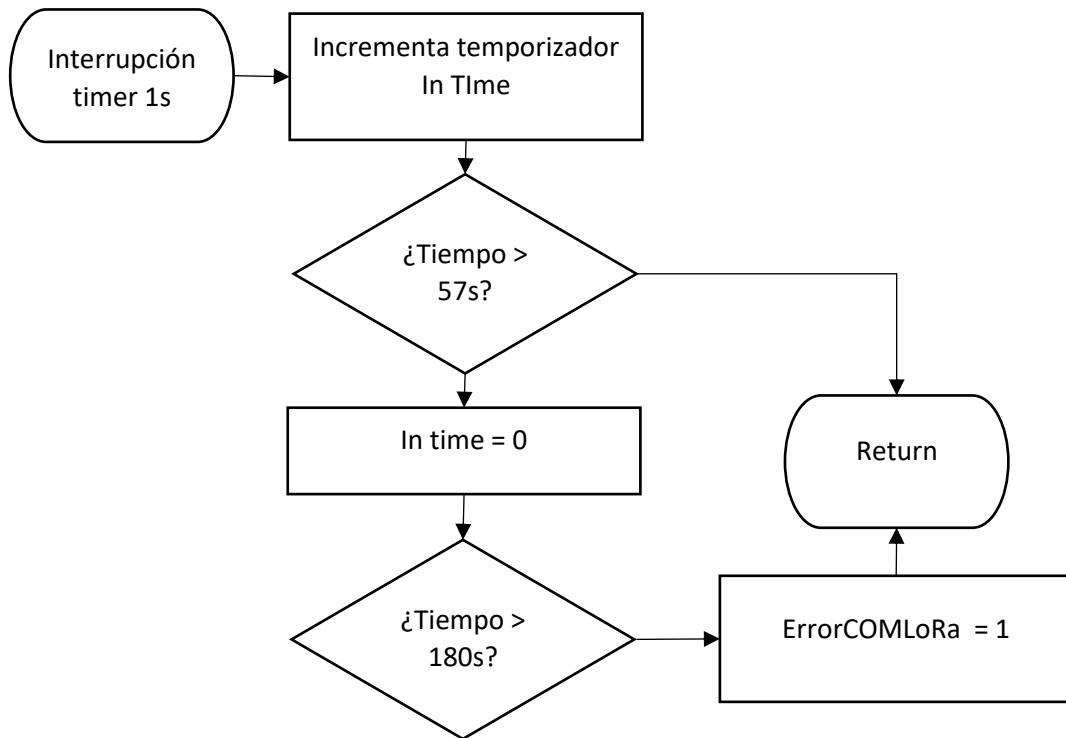


Ilustración 1. 40. Interrupción timer zona vivienda

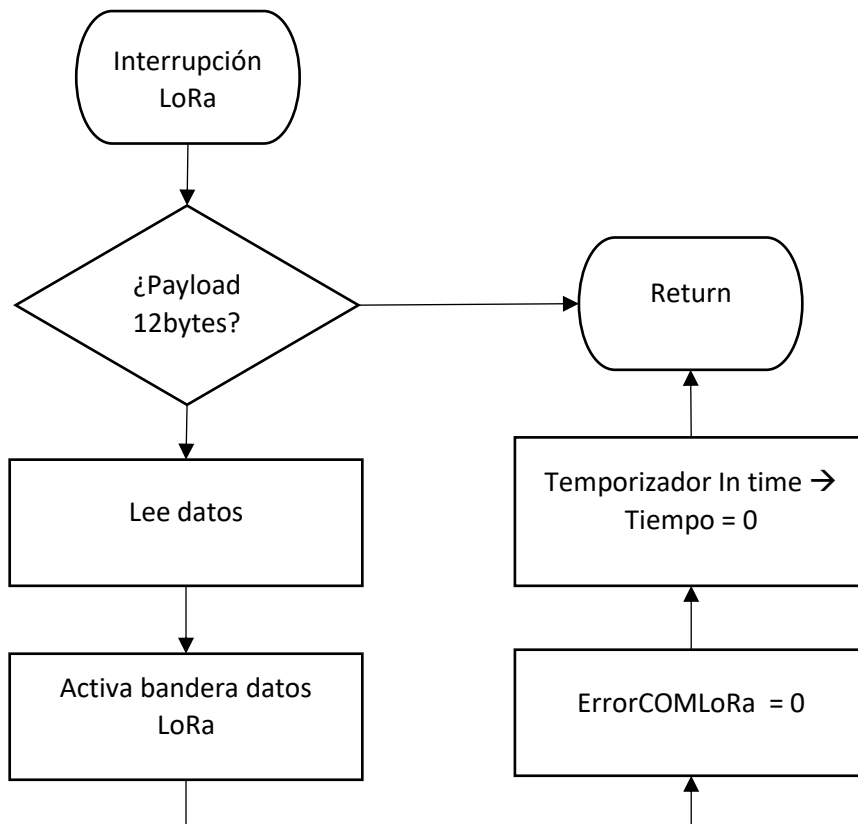


Ilustración 1. 41. Interrupción LoRa zona vivienda

1.8.4. Servidor propio ubicado en Raspberry Pi

El servidor se montará sobre una Raspberry Pi Modelo 3 B+, que cuenta con las siguientes especificaciones:

- **Procesador:** Broadcom BCM2837B0, Cortex-A53 SoC de 64 bits a 1,4 GHz
- **Memoria RAM:** 1GB LPDDR2 SDRAM
- **Conectividad:**
 - WiFi a 2,4 y 5GHz
 - LAN
 - Bluetooth 4.2
 - Gigabit Ethernet sobre USB 2.0 máximo 300Mbps
- **Soporte microSD** donde se aloja el sistema operativo
- Otras especificaciones menos relevantes para la aplicación

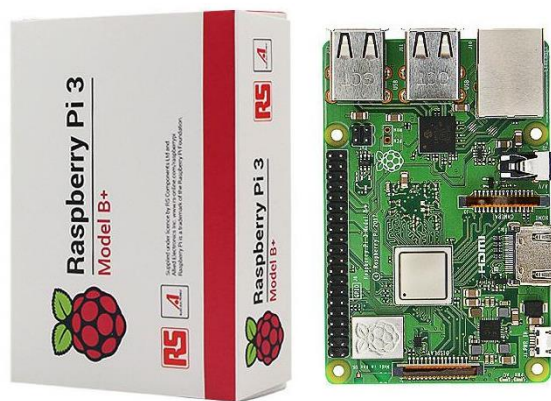


Ilustración 1. 42. Raspberry Pi Model 3B+

Se trata de un hardware modesto, pero suficiente para gestionar una cantidad muy pequeña de usuarios y relativamente pequeña de datos.

1.8.4.1. Tipo de servidor y proceso de instalación

Como ya se ha mencionado en apartados anteriores, se va a montar un servidor LAMP, concretamente con las siguientes versiones:

- Raspbian como distribución de Linux para Raspberry Pi
- Apache como servidor web
- MySQL, como gestor de bases de datos
- PHP como lenguaje de programación

RASPBIAN

Es una distribución de Linux, concretamente Debian, adaptada al hardware reducido de una Raspberry Pi. Existen dos versiones del sistema operativo, con o sin capa gráfica.



Ilustración 1. 43. Logo Debian

En este caso, aunque sería más eficiente utilizar el sistema sin capa gráfica, se ha instalado la versión de escritorio, ya que simplifica la tarea de configuración y acceso a la Raspberry, además de la adaptación a Linux.

Con el sistema de escritorio ocupamos más espacio en la tarjeta SD donde irá instalado el sistema operativo y el sistema irá menos holgado en el funcionamiento, pero reducirá los tiempos de desarrollo del servidor, y en las pruebas se ha obtenido un buen funcionamiento con este sistema.

Raspbian se instala de forma sencilla desde la web oficial de Raspberry Pi, pudiendo descargar directamente el ZIP con la imagen del sistema, o un software de instalación automática.

Versiones Raspbian y kernel:

```
lsb-release -a
Raspbian/GNU/Linux 9.9 (stretch)
uname -a
Linux raspberrypi 4.19.42-v7+ #1219 SMP Tue May 14 21:20:58 BST 2019
armv7l GN
```

INSTALACIÓN DE APACHE, PHP y MYSQL

Con los comandos de Linux en la consola del sistema podemos descargar las versiones necesarias desde los repositorios.

En concreto se instalan los siguientes servicios/aplicaciones:

- Apache2
- PHP 7.0
- Mysql server
- Mysql client
- Php5-mysql

Para realizar la instalación se utilizan los siguientes comandos:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install apache2
sudo apt-get install php7.0 libapache2-mod-php7.0
sudo reboot
sudo apt-get install mysql-server mysql-client
```

CONFIGURAR UNA IP ESTÁTICA PARA EL SERVIDOR

Tener una dirección lógica IP estática (compatible con protocolo IPv4) en el servidor tiene como utilidad poder acceder a ella sabiendo que siempre va a ser la misma.

Si no le asignamos una IP estática el router podría cambiarla si se produce un reinicio o una desconexión, y el código vinculado a esta IP no podría conectarse con el servidor.

Para asignar una IP estática hay que modificar un fichero Linux en el que se almacena esta configuración:

```
/etc/dhcpd.conf
```

Al que podemos acceder desde la consola de linyx utilizando su editor de la siguiente forma:

```
sudo nano /etc/dhcpd.conf
```

y añadir el siguiente código para configurar la conexión LAN y WLAN:

```
interface wlan0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```


La dirección asignada es 192.168.1.100 pero podría ser otra, siempre y cuando sea compatible con la red de clase C privada utilizada 192.168.1.0/24. Se ha elegido esta porque es un valor lo suficientemente alto como para que el router no lo asigne por defecto a otro dispositivo y haya conflictos en la red.

ABRIR EL SERVIDOR A INTERNET

Para abrir el servidor a internet es necesario asignar la IP del servidor al puerto de comunicación 80 utilizado por el protocolo http (número de puerto por defecto utilizado en todos los servidores web sin implementación de seguridad) del router. Se habilita la opción de reenvío de puertos (Port Forwarding) con el objetivo de que todos los paquetes de comunicación que lleguen al router utilizando este puerto, sean redirigidos automáticamente a esta dirección lógica asignada al servidor.

Para ello se accede a la configuración del router al que está conectado el servidor, en el apartado de reenvío de puertos, y creamos una *IPv4*.

Una vez hecho esto comprobamos la IP pública que tiene nuestro router, y a través de ella accederemos al servidor.

Para que esta IP se actualice automáticamente y no sea necesario introducirla se utiliza un servicio de DDNS (Dinamic Domain Name System), que permite actualizar automáticamente la IP pública de nuestro servidor y mantenerla asociada a un dominio en internet.

Para este caso se ha utilizado un servicio gratuito de DDNS llamado Duck DNS.



Ilustración 1. 44. Logo DuckDNS

La configuración de este servicio es muy sencilla, simplemente se crea la cuenta, se añade un dominio y se entra en el apartado install para instalarlo en el sistema operativo que utiliza el servidor.



Ilustración 1. 45. Ejemplo añadir dominio DuckDNS

Este servicio de forma gratuita permite añadir hasta 5 dominios diferentes y únicamente necesitamos uno.

Al crear el dominio habrá que introducir la IP pública que tenga en ese momento y el servicio se encargará de mantenerla actualizada.

CREAR BASE DE DATOS

Además de la instalación del servidor, para simplificar la gestión de la base de datos se ha instalado el paquete PHPMyAdmin.

```
sudo apt-get install php7-0-mysql phpmyadmin
```

Para crear la base de datos se puede acceder a mysql utilizando el usuario root, o podemos crear otro usuario con permisos para gestionar la base de datos. Para crear el nuevo usuario utilizamos los comandos de mysql en Linux:

```
sudo mysql -p -u root
sudo nano /etc/apache2/apache2.conf
CREATE USER 'newuser'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON '.' TO 'newuser'@'%' WITH GRANT OPTION;
```

Una vez creado el usuario ya se puede acceder desde PHPMyAdmin a la base de datos y desde ahí crear las que necesitemos con una interfaz gráfica muy sencilla de utilizar.

Se trata de una interfaz gráfica a través de la cual podemos gestionar la base de datos. Esta interfaz es accesible desde cualquier navegador accediendo a nuestro servidor con la URL configurada para acceder desde internet en apartados anteriores.

También se puede acceder en local utilizando la IP local, o directamente la IP pública en lugar de la URL para acceder desde internet.

```
LOCAL: 192.168.1.100/phpmyadmin/
PÚBLICO: http://subdomain.duckdns.org/phpmyadmin/
```



Ilustración 1. 46. Página de inicio PHPMyAdmin

A través de un inicio de sesión con el nombre del usuario que hayamos configurado en la base de datos y su contraseña, podemos acceder a una interfaz gráfica para gestionarla.

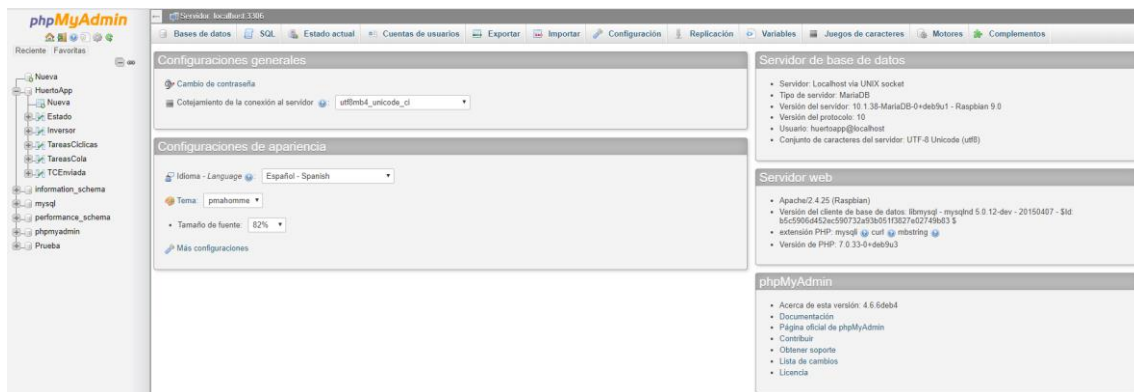


Ilustración 1. 47. Página servidor PHPMyAdmin

Una vez creada la base de datos e instalada la interfaz gráfica para gestionarla más fácilmente, se continua con la creación de las tablas necesarias para almacenar los datos.

Estas tablas se muestran en el siguiente apartado.

1.8.4.2. Tablas de la base de datos

Para el almacenamiento de datos en el servidor se han creado una serie de tablas donde tenemos:

- Datos del inversor
- Estado del sistema
- Tareas para ejecutar
 - Únicas
 - Repetitivas

Es interesante mencionar que, en el servidor, las órdenes de 4 bytes se almacenan en el siguiente formato:

OOOX → Siendo:

- OOO: Tres letras que identifican la orden
- X: Cadena de caracteres que representan el valor en minutos

De forma que el ESP32 al obtener la orden de 4 bytes deberá traducir la X a un valor numérico almacenado en un byte para enviarlo por LoRa.

Se han creado 5 tablas para la gestión del sistema. Estas 5 tablas son las siguientes:

- Tabla de estado. Nombre de la tabla: Estado

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	Id 🗝️	int(11)		No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	FechaHora	timestamp		No	CURRENT_TIMESTAMP		
<input type="checkbox"/>	3	ByteActuadores	tinyint(4)	UNSIGNED	No	Ninguna		
<input type="checkbox"/>	4	WordSensores	smallint(6)	UNSIGNED	No	Ninguna		
<input type="checkbox"/>	5	EstadoCfg	tinyint(3)	UNSIGNED	No	Ninguna		
<input type="checkbox"/>	6	RegandoA	tinyint(1)		No	Ninguna		
<input type="checkbox"/>	7	RegandoB	tinyint(1)		No	Ninguna		
<input type="checkbox"/>	8	Depurando	tinyint(1)		No	Ninguna		
<input type="checkbox"/>	9	LlenandoD	tinyint(1)		No	Ninguna		
<input type="checkbox"/>	10	ErrorCOMLoRa	tinyint(1)		No	Ninguna		

Tabla 1. 14. Tabla Estado

- Tabla de datos del inversor. Nombre de la tabla: Inversor

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	Id 🗝️	int(11)		No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	FechaHora	timestamp		No	CURRENT_TIMESTAMP		
<input type="checkbox"/>	3	P_Aparente	float		No	Ninguna		
<input type="checkbox"/>	4	P_Activa	float		No	Ninguna		
<input type="checkbox"/>	5	P_Bateria	float		No	Ninguna		
<input type="checkbox"/>	6	T_Inversor	float		No	Ninguna		
<input type="checkbox"/>	7	P_Producida	float		No	Ninguna		

Tabla 1. 15. Tabla Inversor

- Tabla de tareas cíclicas. En esta tabla se guardan las tareas programadas semanalmente. Nombre de la tabla: TareasCiclicas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	Id 🗝️	int(11)		No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	Hora	time		No	Ninguna		
<input type="checkbox"/>	3	Dia	tinyint(4)		No	Ninguna	0: Domingo, 6: Sabado	
<input type="checkbox"/>	4	Comando	tinytext	utf8_general_ci	No	Ninguna		
<input type="checkbox"/>	5	Activo	tinyint(1)		No	Ninguna		

Tabla 1. 16. Tabla TareasCiclicas

- Tabla de cola de tareas. En esta tabla se almacenan las tareas pendientes de envío. Nombre de la tabla: TareasCola


#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	Id 			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	FechaHora			No	CURRENT_TIMESTAMP		
<input type="checkbox"/>	3	Comando	utf8_general_ci		No	Ninguna		
<input type="checkbox"/>	4	PendienteEnvio			No	Ninguna		

Tabla 1. 17. Tabla TareasCols

- Tabla de tareas cíclicas enviadas. En esta tabla se almacenan las tareas cíclicas que ya han sido añadidas a la cola de tareas para ser enviadas y evitar que se esté enviando la orden continuamente durante la ejecución. Nombre de la tabla: TCEnviada


#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	Id 			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	Fecha			No	Ninguna		
<input type="checkbox"/>	3	HoraTC			No	Ninguna		

Tabla 1. 18. Tabla TCEnviada

1.8.4.3. API de interacción con el servidor

Para interactuar con la base de datos del servidor se ha desarrollado una API en php que permite, mediante el protocolo HTTP intercambiar información con el servidor de forma práctica.

Se han desarrollado las siguientes *API request*:

- *ClearTask.php*
- *DataInversor.php*
- *NewCtrlStatus.php*
- *NewData.php*
- *NewTask.php*
- *PendingTask.php*
- *ViewStatus.php*
- *Viewtask.php*

Todas utilizan el archivo *HAppConfig.php* para configurar la conexión con la base de datos.

A su vez, todos utilizan una contraseña en uno de los parámetros del GET para evitar ejecutarse sin pasarle ese parámetro extra. Se trata de una pequeña seguridad para los servicios de la API.

Todos los archivos con el código php irán adjuntados en anexos.

- Anexo. [2.7.3. PHP](#)

CLEAR TASK → *ClearTask.php*

Se utiliza para borrar los programas guardados en la base de datos en función de los parámetros de entrada.

Parámetros de entrada:

- 'pwd' → Contraseña de la página php
- 'TASK' → Tipo de tarea:
 - "Riego" → Los programas afectados serán solo los de riego
 - "Depuradora" → Los programas afectados serán solo los de depuradora
- 'DELETE': Instrucción de borrado
 - "Today" → Borra los programas de hoy
 - "All" → Borra todos los programas

Afecta a las tablas de Tareas de la base de datos:

- *TareasCiclicas*
- *TareasCola*
- *TCEnviadas*

DATA INVERSOR → *DataInversor.php*

Devuelve en formato *json* los datos del inversor utilizados para la monitorización en la aplicación Android.

Parámetros de entrada:

- 'pwd' → Contraseña de la página
- 'Day' → Día del mes
- 'Month' → Mes
- 'Year' → Año

Parámetros de salida:

- Archivo *json* con los datos de producción, consumo y batería del inversor

Utiliza las tablas:

- *Inversor*

NEW CTRL STATUS → *NewCtrlStatus.php*

Añade una tarea de control de estado a la cola de tareas a la hora de llamada HTTP. Podría cargar cualquier comando de 3 bytes para añadir en tiempo actual a la lista.

Parámetros de entrada:

- 'pwd' → Contraseña de la página php
- 'CMD' → Comando a añadir a la lista

Afecta a las tablas:

- *TareasCola*

NEW DATA → *NewData.php*

Guarda nuevos datos de medida del inversor y de estado en sus tablas correspondientes.

Parámetros de entrada:

- 'pwd' → Contraseña de la página php
- 'VA' → Consumo, potencia aparente
- 'WC' → Consumo, potencia activa
- 'BA' → Porcentaje de batería
- 'TE' → Temperatura del inversor
- 'WP' → Potencia producida
- 'WSens' → Word de estado de sensores
- 'BAct' → Byte de estado de actuadores
- 'ES' → Byte de estado

Tablas afectadas:

- Inversor
- Estado

NEW TASK → *NewTask.php*

Se utiliza para crear nuevas tareas tanto repetitivas como únicas.

Parámetros de entrada:

- 'pwd' → Contraseña de la página
- 'Tipo' → Tipo de tarea
 - "Unico" → añadirá la nueva tarea a la cola de tareas
 - "X" Número de 7 bits que indica que días se repite → añadirá la nueva tarea a las tareas cíclicas
- 'Day' → Día del mes
- 'Month' → Mes
- 'Year' → Año
- 'H' → Hora
- 'Min' → Minuto
- 'Comando' → Orden a añadir

Tablas afectadas:

- *TareasCiclicas*
- *TareasCola*

PENDING TASK → *PendingTask.php*

Comprueba las tablas que gestionan las tareas y devuelve la siguiente tarea a ejecutar. Si hay una nueva tarea cíclica que enviar la añade a tareas cíclicas enviadas y a la cola de tareas

Una vez hecho esto devuelve la siguiente tarea por orden de fecha y hora que cumpla la condición de ser una fecha y hora anterior a al actual.

En la tabla de tareas pendientes se pondrá a 0 el campo de TareaPendiente en la orden que sea enviada.

Además, actualizará el estado del sistema en el servidor en función de la orden enviada para que se pueda observar en la aplicación.

Parámetros de entrada:

- 'pwd' → Contraseña de la página

Parámetros de salida

- Siguinte orden. Devuelve una cadena de caracteres con la siguiente orden en el formato: OOOX → Siendo:
 - OOO: Tres letras que identifican la orden
 - X: Cadena de caracteres que representan el valor en minutos (si es una orden de 4 bytes) o vacía (si la orden es de 3 bytes)

Tablas afectadas:

- *TareasCiclicas*
- *TareasCola*
- *TCEnviadas*
- *Estado*

VIEW STATUS → *ViewStatus.php*

Devuelve los parámetros de estado en formato *json* para ser mostrados en la pantalla de OTROS CONTROLES en la aplicación.

Parámetros de entrada:

- 'pwd' → Contraseña de la página

Parámetros de salida:

- Archivo *json* con los datos de:
 - Estado del depósito
 - Estado de la presión de agua
 - Estado de la configuración

Utiliza las tablas:

- *Estado*

VIEW TASK → *ViewTask.php*

Se utiliza para leer las tareas programadas y el estado de riego y depuradora en la aplicación Android.

Parámetros de entrada:

- 'pwd' → Contraseña de la página
- 'TASK' → Selecciona que apartado leer
 - "Riego"
 - "Depuradora"

Parámetros de salida:

- Archivo *json* con los datos de:
 - Texto con las programaciones de la tarea seleccionada semanales y únicas
 - Texto con el estado de la tarea seleccionada

Tablas utilizadas:

- *TareasCiclicas*
- *TareasCola*
- *Estado*

1.8.5. Aplicación móvil para Android

1.8.5.1. Entorno gráfico de la interfaz de usuario

Diseño de la aplicación móvil.

La aplicación cuenta con 4 pantallas principales, entre las que se puede alternar desde el menú inferior.

Estas cuatro pantallas son:

- Solar
- Riego
- Piscina
- Otro

1.8.5.1.1. SOLAR

Es la pantalla principal, a la que entra la aplicación tras la pantalla de carga. En ella se muestra la siguiente información sobre el sistema:

- Indicador de fecha actual en la parte superior
- Gráfico comparativo entre producción y consumo (en W)
- Gráfico indicador de la batería
- 2 barras de progreso con indicador de producción y consumo actuales (en W o kW)
- 2 indicadores de la producción y consumo totales (en Wh o kWh)
- Un botón para acceder al historial

Pantallas secundarias de SOLAR:

- Historial



Ilustración 1. 48. Aplicación Android. Solar

PANTALLA HISTORIAL

Al historial se accede pulsando el botón historial dentro de la pantalla solar. En él se muestra una pantalla similar, eliminando los indicadores de producción y consumo instantáneos, y añadiendo un selector de fecha en la parte inferior.

Esta pantalla mantiene los siguientes indicadores de la pantalla solar:

- Gráfico comparativo entre producción y consumo (en W)
- Gráfico indicador de la batería
- 2 indicadores de la producción y consumo totales (en Wh o kWh)

Y añade los siguientes elementos:

- Selector de fecha:
 - Tiene 3 botones y un indicador de la fecha seleccionada.
 - Los botones laterales añaden o quitan un día a la fecha del indicador.
 - El botón central, que es transparente y se ubica sobre el indicador de fecha, abre una pequeña ventana flotante con un calendario para seleccionar la fecha.

1.8.5.1.2. RIEGO

Aquí se muestra información sobre el riego, así como un botón para acceder a los controles disponibles.

La información que muestra es:

- Una cabecera con una imagen auxiliar sobre los lados a regar y el nombre de la ventana
- Una cabecera que puede mostrar la siguiente información:
 - Estado del riego: Que lado se está regando o si no está regando
 - Estado de la bomba del pozo. Si el térmico está activado mostrará un indicador de emergencia en lugar del estado del riego
 - Estado de la comunicación LoRa. Si está activado el error de comunicación Lora en el servidor, lo indicará con un mensaje en la cabecera
- Un texto que indica los riegos programados semanales y los únicos en los próximos 7 días
- Un botón para acceder a la pantalla de menú de riego

Pantallas secundarias de RIEGO:

- Menú riego
- Nuevo programa riego

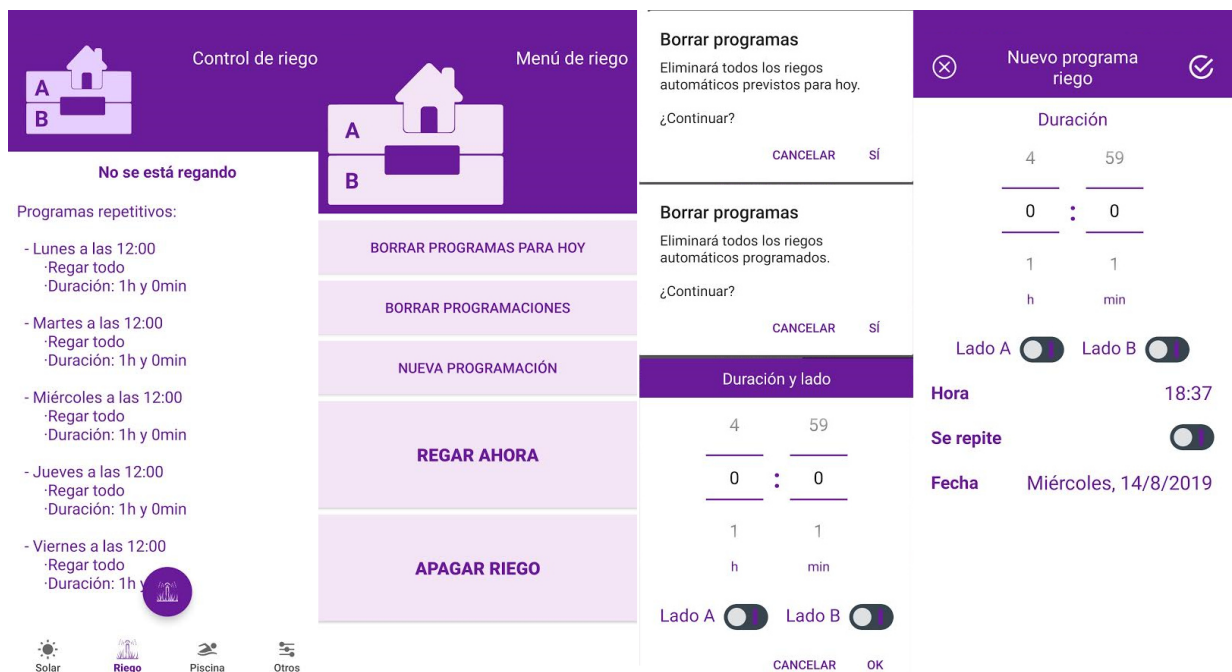


Ilustración 1. 49. Aplicación Android. Riego

MENÚ RIEGO

Tras pulsar el botón de la pantalla de riego se accede a ésta. Aquí se muestran 5 botones que nos permiten realizar diferentes acciones:

- **Borrar programas para hoy:** Este botón borra todos los programas de riego que estén previstos para ejecutarse hoy
- **Borrar programaciones:** Borra por completo todos los programas de riego que haya guardados
- **Nueva programación:** Abre la pantalla de nuevo programa riego
- **Regar ahora:** Abre una ventana flotante en la que se puede seleccionar duración (máximo 4h y 15 minutos, 255 minutos) y lado a regar. En esta ventana flotante se puede iniciar un riego.
- **Apagar riego:** Este botón envía la orden de apagar el riego a la cola de tareas.

NUEVO PROGRAMA RIEGO

A esta pantalla se accede pulsando el botón de nueva programación en MENÚ RIEGO. En esta pantalla se muestra lo siguiente:

- Selector de duración del riego (Máximo 4h y 15 minutos, 255 minutos)
- Selector de lado
- Selector de hora
- Selector Se repite/Único
 - Si se repite mostrará selectores de lunes a domingo para seleccionar los días de la semana
 - Si es único mostrará un selector de fecha
- Botones de validar y cancelar en la parte superior

1.8.5.1.3. PISCINA / DEPURADORA

Esta pantalla tiene un diseño análogo a la de RIEGO, contando con información similar, pero en este caso sobre la depuradora. Cuenta con:

- Una cabecera con una imagen con una funcionalidad estética y el nombre de la pantalla
- Una cabecera que puede mostrar la siguiente información:
 - Estado de la depuradora.
 - Estado de la bomba de la depuradora. Si el térmico está activado mostrará un indicador de emergencia en lugar del estado de la depuradora
 - Estado de la comunicación LoRa. Si está activado el error de comunicación Lora en el servidor, lo indicará con un mensaje en la cabecera
- Un texto que indica los programas de depuración semanales y los únicos en los próximos 7 días
- Un botón para acceder a la pantalla de menú de depuradora

Pantallas secundarias de PISCINA/CONTROL DE DEPURADORA:

- Menú de depuradora
- Nuevo programa depuradora



Ilustración 1. 50. Aplicación Android. Depuradora

MENÚ DEPURADORA

Tras pulsar el botón de la pantalla de riego se accede a ésta. Aquí se muestran 5 botones que nos permiten realizar diferentes acciones:

- **Borrar programas para hoy:** Este botón borra todos los programas de depuradora que estén previstos para ejecutarse hoy
- **Borrar programaciones:** Borra por completo todos los programas de depuradora que haya guardados
- **Nueva programación:** Abre la pantalla de nuevo programa depuradora riego
- **Depurar ahora:** Abre una ventana flotante en la que se puede seleccionar duración (máximo 4h y 15 minutos, 255 minutos). En esta ventana flotante se puede iniciar la depuradora.
- **Apagar depuradora:** Este botón envía la orden de apagar el riego a la cola de tareas.

NUEVO PROGRAMA DEPURADORA

A esta pantalla se accede pulsando el botón de nueva programación en MENÚ DEPURADORA. En esta pantalla se muestra lo siguiente:

- Selector de duración (Máximo 4h y 15 minutos, 255 minutos)
- Selector de hora
- Selector Se repite/Único
 - Si se repite mostrará selectores de lunes a domingo para seleccionar los días de la semana
 - Si es único mostrará un selector de fecha
- Botones de validar y cancelar en la parte superior

1.8.5.1.4. OTROS

Por último, esta pantalla muestra información sobre el estado de otros controles, actuadores y configuraciones.

En la parte superior muestra el estado de diferentes controles. Por el momento muestra 3 estados:

- Nivel del depósito: Lleno, medio o vacío
- Estado del depósito: Llenando, no llenando
- Estado presión: Apagada, Encendida, Error Térmico

Sin embargo, está pensada para mostrar más controles en futuras actualizaciones del sistema.

A su vez muestra 3 botones:

- Control de luces. Esta deshabilitado por el momento, pero en el futuro se tiene pensado añadir un control de las luces de la finca.
- Control de agua. Abre una ventana flotante con 2 botones
 - Llenar/parar llenado depósito
 - Apagar/encender Presión
- Ajustes automáticos. Abre una ventana flotante con los siguientes ajustes:
 - Llenado del depósito automático
 - Presión automática
 - Gestión de excedentes. Por el momento inhabilitado, pero en futuras actualizaciones se espera realizar un sistema de gestión de excedentes en el futuro que depure la piscina o haga otras acciones para aprovecharlos.



Ilustración 1. 51. Aplicación Android. Otros

1.8.5.2. *Conexión con el servidor y funcionamiento de la aplicación*

Para conectar la aplicación con el servidor se hace uso de dos librerías para Android:

- Retrofit
- GSON

Retrofit es una librería que se encarga de realizar las peticiones HTTP con una integración muy buena con GSON.

Por su parte GSON se encarga de enlazar los datos de un archivo *json* a una clase java, simplificando mucho el código necesario para leer datos de un servidor que responde en este formato.

1.8.6. Ideas para mejoras en futuras actualizaciones

En este apartado se proponen una serie de mejoras para estudiar en futuras actualizaciones del sistema. Son ideas generales en las que no se ha profundizado debido a los plazos de desarrollo del proyecto, pero sí que es posible que se implementen más adelante.

1.8.6.1. *Servidor redundante*

Actualmente disponemos de un único servidor ubicado en la vivienda de uno de los propietarios de la finca.

Este servidor está sujeto a caídas de la red de internet o eléctrica de la vivienda en cuestión, así como averías en el mismo. Por lo tanto, sería interesante tener un servidor redundante para tener mayor garantía de funcionamiento del sistema.

En caso de que el error viniera por falta de conexión a internet en la vivienda, el ESP32 allí situado tampoco tendría acceso, por lo que esta solución no cubriría este caso.

Para incluir un servidor adicional podría utilizarse otra Raspberry ubicada en la vivienda de otro de los propietarios, o utilizar un servicio gratuito que cubra requisitos mínimos del sistema, como errores y mediciones recientes.

Este servidor se utilizaría en caso de perder la conexión con el principal, dando mayor garantía de funcionamiento al sistema.

No se trata de una mejora imprescindible, y de hecho, habría que estudiar más en profundidad sobre la rentabilidad de implementarla.

1.8.6.2. *Nuevas Interfaces de usuario*

Sería interesante tener una aplicación web más sencilla que la aplicación Android, para dar cobertura al resto de dispositivos con los que se quiera acceder.

Sin embargo, su uso no iba a ser demasiado, y al igual que con la idea anterior habría que estudiar las ventajas e inconvenientes de incluirla con mayor profundidad.

1.8.6.3. *Copias de seguridad automáticas*

Se ha pensado también en la posibilidad de crear un sistema de copias de seguridad de la base de datos de forma automática, para en caso de fallo del servidor, poder recuperar esos datos.

Los datos de la copia podrían almacenarse en una memoria externa conectada al servidor, o en el servidor redundante en caso de implementarse.

1.8.6.4. Mejoras de la gestión de la base de datos

En un primer momento los datos almacenados no van a suponer una gran carga del servidor en las consultas, pero es probable que, a largo plazo, cuando haya almacenados muchos datos en el mismo se ralenticen las búsquedas.

Por ello se podría mejorar la base de datos con la creación automática de las tablas por meses, en lugar de utilizar una única tabla para todo. De este modo las tablas serían más pequeñas y acceder a ella más ágil.

Esta mejora seguramente se implemente en próximas actualizaciones, ya que únicamente requiere modificar la API del servidor.

1.8.6.5. Mejoras de la aplicación Android

Actualmente el error de comunicación LoRa solo puede mirarse desde las pantallas de riego y depuración, se plantea como mejora mostrar esa información de forma óptima.

Además, también tenemos almacenadas medidas de temperatura del inversor que actualmente no se están mostrando, estaría bien incluir una gráfica para estas medidas.

También, como se ha mencionado en el apartado de la interfaz de usuario, hay mejoras que están implementadas en la interfaz de usuario, pero no en el sistema, como son el control de luces y la gestión de excedentes. Estas mejoras requerirán también de alguna pequeña modificación en la aplicación.

Se han pensado a su vez mejoras en la forma de borrar las programaciones realizadas para el riego y la depuradora. Estas mejoras incluirían poder borrar un programa en concreto en lugar de tener que borrar los del día actual o todas como únicas opciones.

1.8.6.6. Mejoras del programa de control

En el programa de control habrá que incluir también las mejoras de gestión de excedentes y control de iluminación mencionado anteriormente.

Además, podría incluirse mayor seguridad en el sistema teniendo en cuenta los estados de alarma que envía el inversor por la trama y actualmente no se leen. También se podría tener en cuenta el valor de temperatura para reducir consumos en caso de temperaturas elevadas y añadir una protección extra a la red, evitando que sea el inversor el que corte el suministro.

1.8.6.7. *Mejor gestión del consumo de agua*

Para poder llevar un control de la cantidad de agua que se extrae del pozo en la primera versión se podrá consultar en la base de datos las horas a la que ha estado activada, con consultas SQL.

La idea es añadir un control de las horas que ha estado activada en la aplicación Android para el facilitar el control.

De este modo se podrá cumplir más fácilmente con las limitaciones legales de extracción de agua.

1.9. Planificación

Etapas

- Investigación
 - Tecnología LPWAN
 - Almacenamiento de datos (servidor)
 - Hardware LPWAN
- Selección del hardware necesario
- Implementación del servidor
 - Instalación del servidor
 - Crear base de datos
 - Abrir el servidor a internet
 - Desarrollar API
- Desarrollo del software de control y comunicación
 - Software de control
 - Protocolo de comunicaciones
 - Conexión con el servidor
- Desarrollo de la aplicación para Android
 - Interfaz gráfica
 - Conexión con el servidor
- Fase de pruebas y corrección de errores con el simulador
- Fase de pruebas en montaje real
- Funcionamiento comprobado, finalización del proyecto

INVESTIGACIÓN

El primer proceso por realizar es investigar sobre las distintas tecnologías que se van a utilizar para desarrollar el proyecto.

A esta tarea se le asigna una semana de trabajo.

- 2 días para investigar sobre tecnologías LPWAN y el hardware compatible
- 1 día para el hardware necesario para controlar el sistema
- 2 días para la investigación sobre servidores y almacenamiento de datos en la nube

Tiempo asignado: 5 días

Horas diarias: 8h

Horas totales de esta fase: 40h

SELECCIÓN DEL HARDWARE

Una vez estudiadas las posibilidades hay que seleccionar el hardware que se va a utilizar con las tecnologías que se hayan decidido.

- 1 día para la selección del hardware necesario para el control
- 1 día para el hardware necesario para las comunicaciones y el almacenamiento de datos.

Tiempo asignado: 2 días

Horas diarias: 8h

Horas totales de esta fase: 16h

IMPLEMENTACIÓN DEL SERVIDOR

Fase de implementación del servidor y desarrollo de la API para almacenar y consultar los datos.

- 1 día para:
 - Instalación y configuración del servidor
 - Creación de la base de datos
 - Abrir el servidor a internet

- 4 días para desarrollar API

Tiempo asignado: 5 días

Horas diarias: 8h

Horas totales de esta fase: 40h

SOFTWARE DE CONTROL Y COMUNICACIÓN

- 2 días para el software de control en la finca
- 1 días para el protocolo de comunicaciones
- 1 día para el software de comunicación con el servidor
- 1 día para unir el software para ESP32 de los tres apartados mencionados

Tiempo asignado: 5 días

Horas diarias: 8h

Horas totales de esta fase: 40h

DESARROLLO DE LA APLICACIÓN ANDROID

Etapas de desarrollo de la aplicación para la interfaz de usuario.

- 4 días para el desarrollo de la interfaz gráfica
- 3 días para la programación de interacción de usuario con la interfaz gráfica
- 1 día para el desarrollo de tratamiento de datos y conexión con el servidor

Tiempo asignado: 8 días

Horas diarias: 8h

Horas totales de esta fase: 64h

1.10. Orden de prioridad entre los documentos

Se mantiene el orden de prioridad de los documentos por defecto:

- 1. Planos
- 2. Pliego de condiciones
- 3. Presupuesto
- 4. Memoria

CAPÍTULO 2. ANEXOS

ÍNDICE DE ANEXOS

ÍNDICE DE ANEXOS	87
ÍNDICE DE ILUSTRACIONES.....	90
2.1. Documentación de partida.....	91
2.2. Cálculos	92
2.3. Estudio de tecnologías LPWAN	94
2.3.1. Sigfox	94
2.3.2. LoRa	95
2.3.3. LoRaWAN.....	96
2.4. Prueba de comunicación real entre la finca y la vivienda	97
2.5. Prueba de comunicación real con el inversor	98
2.6. Modelo de simulación	99
2.6.1. Maqueta	99
2.6.2. Software simulador del inversor	99
2.7. Código desarrollado	102
2.7.1. Android Studio.....	102
2.7.1.1. XML Layout.....	102
2.7.1.1.1. activity_solar	102
2.7.1.1.2. activity_historial_dia	107
2.7.1.1.3. activity_riego.....	110
2.7.1.1.4. activity_menuriego	113
2.7.1.1.5. activity_prog_riego	116
2.7.1.1.6. activity_piscina.....	124
2.7.1.1.7. activity_menudep	127
2.7.1.1.8. activity_prog_dep	129
2.7.1.1.9. activity_otros	137
2.7.1.1.10. dialog_ajustes_auto	141
2.7.1.1.11. dialog_calendar.....	143
2.7.1.1.12. dialog_ctrl_agua.....	144
2.7.1.1.13. dialog_duracion.....	145
2.7.1.1.14. dialog_reg_ahora	147

2.7.1.1.15.	dialog_time	149
2.7.1.2.	Código Java	150
2.7.1.2.1.	Activities	150
2.7.1.2.1.1.	SolarActivity	150
2.7.1.2.1.2.	HistorialDiaActivity	154
2.7.1.2.1.3.	RiegoActivity	158
2.7.1.2.1.4.	MenuRiegoActivity	161
2.7.1.2.1.5.	ProgRiegoActivity	165
2.7.1.2.1.6.	PiscinaActivity	169
2.7.1.2.1.7.	MenuDepActivity	172
2.7.1.2.1.8.	ProgDepActivity	176
2.7.1.2.1.9.	OtrosActivity	180
2.7.1.2.2.	ChartModels	185
2.7.1.2.2.1.	LineChartBatery	185
2.7.1.2.2.2.	LineChartPower	189
2.7.1.2.3.	FnModels	194
2.7.1.2.3.1.	FormatoFecha	194
2.7.1.2.3.2.	GridLines	196
2.7.1.2.4.	RasPiServer	197
2.7.1.2.4.1.	RasPiService	197
2.7.1.2.4.2.	RasPi	198
2.7.1.2.5.	ServerModels	199
2.7.1.2.5.1.	RasPiCtrlStat	199
2.7.1.2.5.2.	RasPiData	200
2.7.1.2.5.3.	RasPiInversor	201
2.7.1.2.5.4.	RasPiMonitorInv	203
2.7.1.2.5.5.	RasPiTask	204
2.7.1.2.5.6.	RasPiValores	205
2.7.2.	ESP32	206
2.7.2.1.	Zona finca	206
2.7.2.1.1.	Archivo principal	207
2.7.2.1.2.	00_Setup	209
2.7.2.1.3.	01_FnPrincipal	211
2.7.2.1.4.	02_Fn_Comprueba_Maquinas	212
2.7.2.1.5.	03_Maquinas	213

2.7.2.1.6.	04_Fn_RiegoDeposito	214
2.7.2.1.7.	05_Fn_Depuradora	216
2.7.2.1.8.	06_Fn_Presion.....	216
2.7.2.1.9.	07_Interrupciones.....	217
2.7.2.1.10.	08_Fn_Datos_INT.....	218
2.7.2.1.11.	09_Fn_Orden:INT.....	220
2.7.2.2.	Zona vivienda	221
2.7.2.2.1.	Principal.....	222
2.7.2.2.2.	Funciones	223
2.7.2.2.3.	Funciones_WiFi.....	224
2.7.2.2.4.	Interrupción_Timer	225
2.7.2.2.5.	LoRa.....	225
2.7.2.2.6.	Setup	226
2.7.3.	PHP	228
2.7.3.1.	HAppConfig.....	229
2.7.3.2.	ClearTask	230
2.7.3.3.	DataInversor	231
2.7.3.4.	NewCtrlStatus.....	233
2.7.3.5.	NewData.....	234
2.7.3.6.	NewTask	235
2.7.3.7.	PendingTask	236
2.7.3.8.	ViewStatus.....	238
2.7.3.9.	ViewTask.....	239

ÍNDICE DE ILUSTRACIONES

Ilustración 2. 1. Programa de cálculo LoRa SX1276	93
Ilustración 2. 2. Logo Sigfox.....	94
Ilustración 2. 3. Logo LoRa.....	95
Ilustración 2. 4. Ejemplo LoRaWAN.....	96
Ilustración 2. 5. Prueba de comunicación con el inversor	98
Ilustración 2. 6. Maqueta de simulación	99
Ilustración 2. 7. Interfaz de usuario simulador inversor	100
Ilustración 2. 8. Código LabView simuldaor inversor	101

2.1. Documentación de partida

Como documentación de partida se dispone de los siguientes manuales:

- Manual de Semtech para la configuración LoRa
- Manual inversor

Así como toda la información sobre los módulos y componentes utilizados.

Todas las referencias a esta documentación están vinculadas en la bibliografía para poder acceder a ella.

- Bibliografía. [1.4.4. Bibliografía](#)

2.2. Cálculos

Para este proyecto únicamente ha sido necesario realizar cálculos sobre el tiempo de duración de los mensajes LoRa en el aire y así cumplir la normativa con el protocolo utilizado.

Para estos cálculos se ha utilizado el programa proporcionado por Semtech: “LoRa Modem Calculator Tool”.

En el programa se introducen los siguientes parámetros:

- Spreading factor
- Bandwidth
- Coding rate
- Preamble
- Payload

La configuración elegida para cumplir con los requisitos planteados es la siguiente:

- **Spreading factor o tiempo de esparcimiento.** Se ha seleccionado en 11, siendo su valor máximo 12 y el mínimo 6. Cuanto mayor es este valor más tiempo dura el mensaje en el aire, haciéndolo más robusto a interferencias, pero aumentando también el tiempo necesario a dejar entre mensajes. Es un factor de escala que afecta a todo el mensaje.
- **Bandwidth.** Ancho de banda. Cuanto menor, más lenta es la comunicación. Siguiendo el manual de diseño de Semtech es conveniente elegir los 125kHz para factores de esparcimiento altos.
- **Coding rate.** Sistema para corregir errores de comunicación, se puede configurar entre 1 y 4 entre, afectando a un divisor, por lo que con un valor de 1 tenemos mayor coding rate. Para el caso se ha seleccionado 1.
- **Preamble.** Preámbulo previo al mensaje principal. Se utiliza principalmente en redes LoRaWAN, y su valor mínimo es 2. La configuración se ha dejado como estaba por defecto en 6, dejando un margen mayor de maniobra.
- **Payload.** Duración del mensaje. Se ha establecido en 12 bytes por los motivos mencionados en el diseño del protocolo.

Los resultados que nos muestra el programa son los siguientes:

The screenshot shows the 'LoRa Modem Calculator Tool' interface. It is divided into several sections:

- Calculator Inputs:**
 - LoRa Modem Settings:** Spreading Factor (11), Bandwidth (125 kHz), Coding Rate (1), Low Datarate (Optimiser On).
 - Packet Configuration:** Payload Length (12 Bytes), Programmed Preamble (6 Symbols), Total Preamble Length (10.25 Symbols), Header Mode (Explicit Header Enabled), CRC Enabled (Enabled).
 - RF Settings:** Centre Frequency (865000000 Hz), Transmit Power (17 dBm), Hardware Implementation (RFIO is Shared).
 - Compatible SX Product:** 1272, 1276
- Selected Configuration:** A schematic diagram showing the VR_PA, RFO, and RFI components connected to Tx and Rx ports. Below the diagram are three boxes labeled 'Preamble', 'Payload', and 'CRC'.
- Calculator Outputs:**
 - Timing Performance:** Equivalent Bitrate (537.11 bps), Time on Air (462.85 ms), Preamble Duration (167.94 ms), Symbol Time (16.38 ms).
 - RF Performance:** Link Budget (151.5 dB), Receiver Sensitivity (-134.5 dBm), Max Crystal Offset (36.1 ppm).
 - Consumption:** Transmit (90 mA), CAD/Rx (10.8 mA), Sleep (100 nA).

At the bottom, a summary line reads: SF = 11, BW = 125 kHz, CR = 4/5, Header Disabled, Preamble = 10.25 syms, Payload = 12 bytes.

Ilustración 2. 1. Programa de cálculo LoRa SX1276

Con esta configuración obtendríamos un tiempo en el aire del mensaje de 462,85ms, con lo que, sabiendo que este tiempo es el 1%, habrá que dejar 100 veces ese valor entre mensajes.

De modo que habrá que dejar como mínimo:

46,285s entre mensajes.

Con este dato podemos calcular el número máximo de mensajes al día:

$$1 \text{ día} = 24h * 60 \text{ min} * 60s = 86400s$$

$$N^{\circ} \text{ de mensajes/día} = \frac{86400}{46,285} = 1866 \text{ mensajes/día}$$

2.3. Estudio de tecnologías LPWAN

Este tipo de redes se caracteriza por tener un consumo energético muy bajo y una capacidad de comunicación de larga distancia (incluso kilómetros) a costa de tener unas velocidades de transmisión bajas.

Son un tipo de red muy utilizado en IoT debido a su bajo consumo, unido a la utilización de bandas libres de radio para la comunicación, resultando muy económicas y sencillas de mantener.

Las redes de este tipo más utilizadas son Sigfox, LoRa y LoRaWAN, por lo que vamos a estudiar las características de las tres.

2.3.1. Sigfox

Se trata de una compañía francesa, que ofrece una solución a las redes de comunicación de IoT actuando como proveedor de red.



Ilustración 2. 2. Logo Sigfox

La compañía utiliza bandas de frecuencia libre de 868MHz en Europa y 902MHz en Estados Unidos para proporcionar una red Ultra-NarrowBand en la que se sostiene su servicio.

Al utilizar una red propia es necesario adquirir un hardware de radio que sea compatible y contratar una suscripción del servicio a la compañía. La compañía dispone de cobertura en la mayoría de los países de Europa occidental, entre los que se encuentra España.

Las limitaciones de la red son:

- Se pueden enviar 140 mensajes al día con un límite de 7 mensajes cada hora
- Cada mensaje es una trama de hasta 12 bytes
- El módulo puede recibir 4 transmisiones por día

Podemos ver como esta red, a pesar de ser interesante para IoT, podía adaptarse a la monitorización del sistema, pero al no poder recibir apenas mensajes hace prácticamente imposible el control de los actuadores a través de este medio.

Al tener estas limitaciones para el diseño no cumple los requisitos del sistema en su totalidad, aunque podría ser utilizada parcialmente sólo para la monitorización, siendo necesario otro sistema para el control. Por otro lado, al disponer de su propia red, SigFox nos solucionaría el problema del servidor para almacenar los datos.

Otras características de la red son:

- La compañía gestiona por completo la comunicación, haciendo que la integración del módulo sea muy sencilla, reduciendo el tiempo de desarrollo.
- SigFox proporciona una única API para interactuar con el módulo.
- No requiere ninguna configuración
- Es compatible con Bluetooth, GPS, 2G, 3G, 4Gy WiFi

2.3.2. LoRa

LoRa es un sistema de modulación de radiofrecuencia con una gran capacidad para realizar conexiones seguras a grandes distancias en redes de Internet de las Cosas, donde se necesitan dispositivos de bajo consumo.



Ilustración 2. 3. Logo LoRa

Este sistema está patentado por la empresa Semtech que se dedica a la manufacturación de chips de comunicaciones por radio. Sin embargo, en la actualidad, esta tecnología se administra a través de LoRa Alliance.

Esta tecnología cuenta con las siguientes ventajas:

- Soporta muy bien las interferencias
- Tiene una sensibilidad alta a la hora de recibir datos
- Se basa en el sistema de modulación “chirp” (frecuencia modulada pulsada)
- Larga durabilidad de la batería debido al bajo consumo
- Alcance de hasta 20km
- Conexión bidireccional punto a punto
- Utiliza bandas libres de frecuencia (868MHz Europa, 433MHz Asia, 915MHz América)

Como desventaja podría destacarse que se dispone de una velocidad de transmisión baja, por lo que no es adecuada para la transferencia de gran cantidad de datos en poco tiempo. Entre sus limitaciones están a parte de la velocidad, la banda de frecuencia que utilizemos y la legislación vigente para cada zona del mundo.

Sin embargo, para la aplicación indicada cumple todos los requisitos, siendo además una plataforma libre, con la que se puede montar una red independiente de empresas de terceros.

Esta tecnología es ideal para esta clase de sistemas, en los que hay mala o ninguna conexión a internet y se desea crear una red privada de sensores y actuadores.

2.3.3. LoRaWAN

Basado en la tecnología LoRa, tenemos esta especificación para redes Low Power Wide Area Network (LPWAN) que se compone de nodos y gateways.

El Gateway es el encargado de comunicar con los nodos y enviarles y recibir información del servidor. Por otro lado, los nodos son los dispositivos últimos, donde se encuentran por ejemplo los sensores, encargados de comunicarse con su Gateway correspondiente.

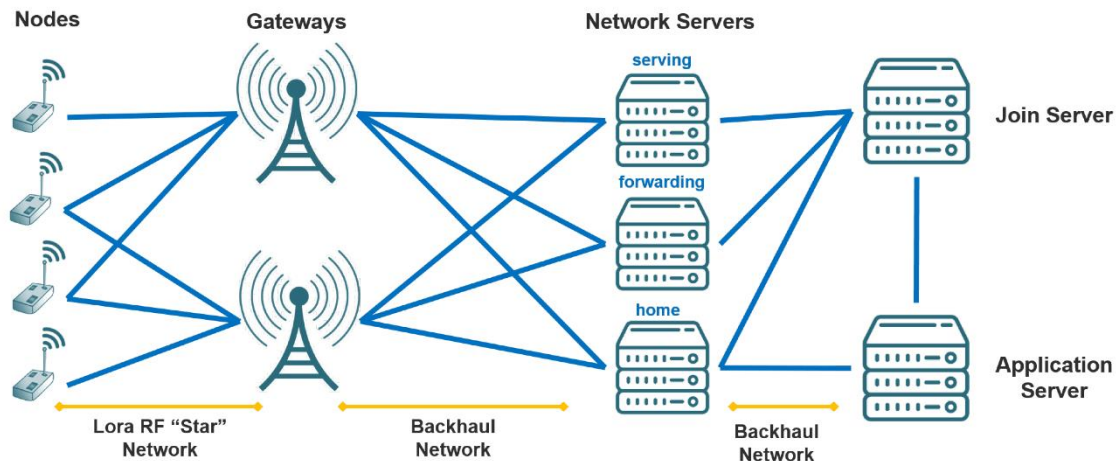


Ilustración 2. 4. Ejemplo LoRaWAN

Este tipo de red podría servir para la aplicación en cuestión, ya que cuenta con las ventajas de LoRa y además existen ya módulos diseñados para este tipo de redes.

Sin embargo, realizar con este método el proyecto sería más caro que utilizar una red LoRa diseñada específicamente para este sistema, ya que requiere de menos componentes, y al tratarse de un sistema con pocos nodos y que solo necesitaría un Gateway, las ventajas de utilizar este sistema no serían apreciables.

Esto se debe a que la principal ventaja sería que la estructura de la red ya estaría diseñada, pero al requerirse de pocos nodos esta ventaja es pequeña respecto a un diseño propio.

Por otro lado, un diseño propio permite mayor flexibilidad y eficiencia.

2.4. Prueba de comunicación real entre la finca y la vivienda

Se realizó una prueba de la comunicación real para comprobar si cumplía con la distancia a la que se requería conectar.

Esta prueba consistía en cargar un programa de ejemplo de la librería de comunicación LoRa en dos módulos y situar uno en la finca y otro en la vivienda, comprobando que se establecía la comunicación.

En las pruebas se comprobó que se conseguía comunicar en prácticamente toda la parcela de la finca, exceptuando algunas zonas con mucha densidad de árboles.

Con estas pruebas se definió el punto de instalación de los módulos en los dos puntos de comunicación.

En la vivienda se coloca en una terraza con visibilidad directa con la finca, mientras que en la finca se coloca en la zona próxima al inversor y el cuadro eléctrico, para poder realizar las conexiones de forma más sencilla.

En estos dos puntos la comunicación es satisfactoria y resulta práctico ubicarlos de esa manera.

2.5. Prueba de comunicación real con el inversor

Para probar la comunicación con el inversor se conectó vía puerto serie a los pines Serial2 del ESP32 con la interfaz de TTL a RS232 que aportaba el chip MAX3232 integrado en la placa adquirida para adaptar la comunicación.

El script utilizado envía el comando *QPIGS<cr>* a través de la UART2 que comunica con el inversor, espera a recibir la respuesta, y la imprime en el puerto serie 1, que está conectado al PC.

De este modo, monitorizando el puerto serie podemos ver la trama que devuelve. Algunas de las muestras de prueba de la trama tomadas son las siguientes:

```
(000.0 00.0 230.1 49.9 1680 1662 033 354 50.80 000 086 0047 0014 082.8
50.88 00021 00110110 00 00 00712 010"u
(000.0 00.0 230.0 50.0 1679 1672 033 353 50.80 000 086 0047 0013 088.3
50.80 00021 00110110 00 00 00706 010??
(000.0 00.0 230.0 50.0 1679 1675 033 352 50.60 000 084 0047 0014 086.4
50.57 00021 00110110 00 00 00713 010??
(000.0 00.0 230.0 49.9 1679 1668 033 351 50.50 000 083 0047 0013 088.9
50.57 00021 00110110 00 00 00702 010TZ
(000.0 00.0 230.1 49.9 1679 1668 033 350 50.30 000 081 0047 0014 084.6
50.41 00021 00110110 00 00 00715 010??
(000.0 00.0 229.9 50.0 1678 1671 033 350 50.30 000 081 0047 0014 082.1
50.33 00021 00110110 00 00 00709 010a
(000.0 00.0 230.0 50.0 1679 1668 033 349 50.20 000 080 0047 0014 084.0
50.17 00021 00110110 00 00 00732 010??
(000.0 00.0 230.1 50.0 1679 1668 033 349 50.10 000 079 0047 0014 081.5
50.17 00021 00110110 00 00 00722 010'?
(000.0 00.0 230.1 50.0 1679 1666 033 348 50.00 000 078 0047 0014 085.4
50.02 00021 00110110 00 00 00745 010){
```

Ilustración 2. 5. Prueba de comunicación con el inversor

2.6. Modelo de simulación

2.6.1. Maqueta

Se ha montado una pequeña maqueta para llevar a cabo la simulación de funcionamiento del sistema.

Mediante unos leds podemos simular el estado de los actuadores y con unas palancas el de los sensores digitales.

La maqueta consta de:

- 3 leds verdes que representan las 3 bombas encendidas
- 3 leds azules que representan las 3 electroválvulas abiertas
- 2 leds amarillos que representan los sensores de nivel del depósito con 2 palancas para controlarlos.
- 1 led blanco que representa el estado de conexión wifi en la vivienda.

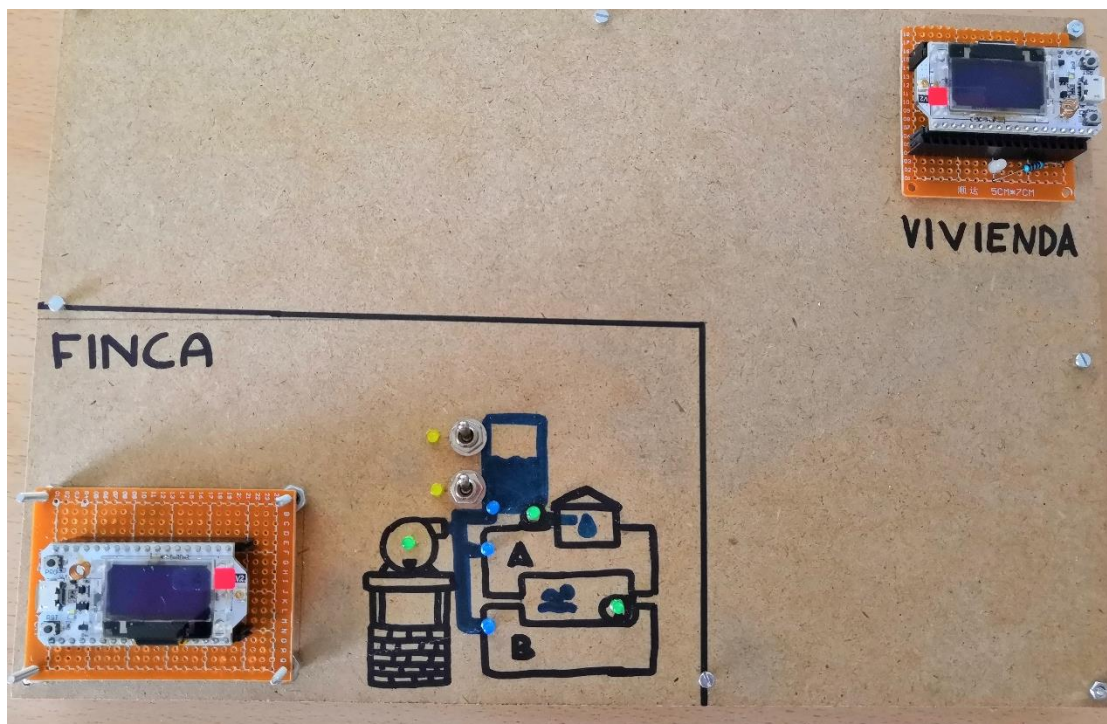


Ilustración 2. 6. Maqueta de simulación

2.6.2. Software simulador del inversor

El software desarrollado para simular el inversor es una pequeña aplicación en LabView que simula el protocolo de comunicaciones que sigue el inversor.

El inversor se comunica vía puerto serie a 2400 baudios. En el proyecto solo vamos a enviar un tipo de comando, que nos responde con el estado de producción y consumo, así como información sobre temperatura y carga de las baterías.

En la misma trama responde una serie de datos adicionales que pueden consultarse en el manual, pero que no vamos a tener en cuenta, ya que muchos son datos sobre la configuración del inversor que ya conocemos.

La trama tiene el siguiente formato:

**(BBB.B CCC.C DDD.D EE.E FFFF GGGG HHH III JJ.JJ KKK OOO TTTT EEEE UUU.U WW.WW
PPPPP 76543210 UU UU XXXXX 010TZ**

Cada grupo de caracteres se corresponde con un dato de información del sensor. Los datos que interesan para el proyecto son:

- FFFF → Valor de consumo en potencia aparente (VA)
- GGGG → Valor de consumo en potencia activa (W)
- OOO → Porcentaje de batería
- TTTT → Temperatura del inversor
- XXXXX → Valor de producción en W

Por lo tanto, en el simulador solo se modificará esa parte de la trama, ya que el resto se leerá y es indiferente que responda letras.

Se desarrolló la siguiente interfaz gráfica:



Ilustración 2. 7. Interfaz de usuario simulador inversor

Por su parte, el código es bastante sencillo y puede verse en la siguiente captura de pantalla del diagrama de bloques en el entorno de programación de LabView:

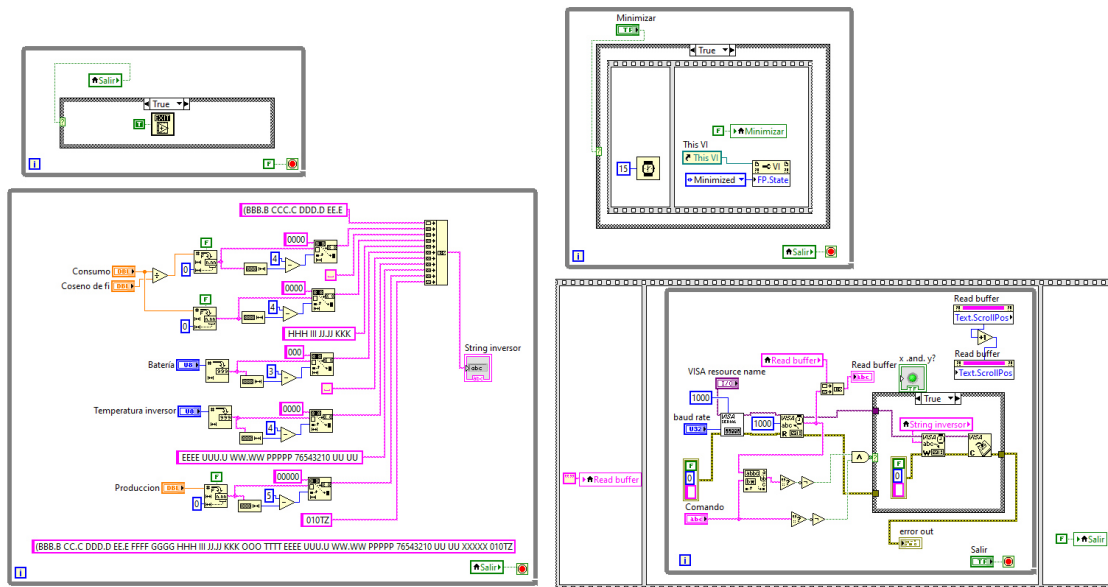


Ilustración 2. 8. Código LabView simulador inductor

Mediante este software podemos conectar a un PC el ESP32 con un cable USB y simular la comunicación serie con el inductor.

Este software ahorra tiempo de desarrollo permitiendo realizar pruebas al mismo tiempo que se realiza el software de control del sistema y poder simular situaciones de comportamiento con pruebas en función de las medidas que nosotros queramos.

2.7. Código desarrollado

El código desarrollado se adjunta a continuación.

2.7.1. Android Studio

2.7.1.1. XML Layout

2.7.1.1.1. activity_solar

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.SolarActivity">

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@color/colorPrimaryDark"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0">

    <Button
        android:id="@+id/button_A_Solar"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:layout_weight="1"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_sun"
        android:drawableTint="@color/colorBTN_ON"
        android:text="@string/ButtonA"
        android:textColor="@color/colorBTN_ON"
        android:background="@color/colorBTN_BG"
        android:textStyle="bold" />

    <Button
        android:id="@+id/button_B_Solar"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:layout_weight="1"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_riego"
        android:drawableTint="@color/colorBTN_OFF"
        android:text="@string/ButtonB"
        android:textColor="@color/colorBTN_OFF"
        android:background="@color/colorBTN_BG"
        />
</LinearLayout>
</ConstraintLayout>
```

```

<Button
    android:id="@+id/button_C_Solar"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:layout_weight="1"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_piscina"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonC"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
/>

<Button
    android:id="@+id/button_D_Solar"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:layout_weight="1"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_control3"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonD"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:id="@+id/TopSolar"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/ScrollViewSolar"
    >
    <LinearLayout
        android:id="@+id/backgroundTOPSolar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        android:orientation="horizontal"
        android:background="@color/colorPrimaryDark"
        >
        <TextView
            android:id="@+id/TVFechaSolar"
            android:layout_width="match_parent"
            android:layout_height="30dp"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="10dp"
            android:background="@color/colorPrimaryDark"
            android:textColor="@color/colorWhite"
            android:text=""
            android:textSize="20sp"
            android:textAlignment="textEnd"
            />
    </LinearLayout>
</LinearLayout>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="205dp"
    android:background="@color/colorPrimaryDark">

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/LCPowerSolar"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginTop="0dp"
        android:background="@color/colorPrimaryDark" />

</RelativeLayout>

</LinearLayout>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:id="@+id/ScrollViewSolar"
    android:background="@color/colorWhite"
    app:layout_constraintBottom_toTopOf="@+id/LinearLayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/TopSolar"
    app:layout_constraintVertical_bias="0.0">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/LinearLayoutSolar"
        android:background="@color/colorWhite">

        <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TextView
                android:id="@+id/BatActualSolar"
                android:text="0 %"
                android:layout_alignParentEnd="true"
                android:layout_marginEnd="10dp"
                android:layout_marginTop="5dp"
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:drawableEnd="@drawable/bat_ico"
                android:textSize="20sp"
                android:textStyle="bold"
                android:textAlignment="center"
                android:textColor="@color/colorPrimaryDark"
            />

            <com.github.mikephil.charting.charts.LineChart
                android:id="@+id/LCBatterySolar"
                android:layout_marginBottom="0dp"
                android:layout_marginTop="8dp"
                android:layout_width="match_parent"
                android:background="@color/Transparent"
                android:layout_height="180dp"/>

        </RelativeLayout>

    </LinearLayout>

</ScrollView>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="150dp">

    <ProgressBar
        android:id="@+id/bar1"
        style="?android:attr/progressBarStyleHorizontal"

```

```

        android:layout_width="160dp"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:max="5000"
        android:progress="5000"
        android:progressTint="@color/ColorProduccion"
        android:backgroundTint="@color/colorPrimaryDark"
        android:progressDrawable="@drawable/circularprogress"
        android:background="@drawable/circularprogressbg"
        android:layout_marginStart="25dp"
    />

    <TextView
        android:id="@+id/ProdInst"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_centerVertical="true"
        android:text="W"
        android:textSize="18sp"
        android:textStyle="bold"
        android:textColor="@color/ColorProduccion"
        android:textAlignment="center"
    />

    <ProgressBar
        android:id="@+id/barCons"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="160dp"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:max="5000"
        android:progress="5000"
        android:progressTint="@color/ColorConsumo"
        android:backgroundTint="@color/colorPrimaryDark"
        android:progressDrawable="@drawable/circularprogress"
        android:background="@drawable/circularprogressbg"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="25dp"
    />

    <TextView
        android:id="@+id/ConsumoInst"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="25dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:text="W"
        android:textSize="18sp"
        android:textStyle="bold"
        android:textColor="@color/ColorConsumo"
        android:textAlignment="center"
    />

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/ProdActualSolar"
        android:text="0 Wh"
        android:layout_marginStart="25dp"
        android:layout_centerVertical="true"
        android:layout_height="wrap_content"
        android:layout_width="160dp"
        android:drawableBottom="@drawable/prod_ico"
        android:textSize="25sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"
    />

```

```

<TextView
    android:id="@+id/ConsumoActualSolar"
    android:text="0 Wh"
    android:layout_marginEnd="25dp"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:layout_height="wrap_content"
    android:layout_width="160dp"
    android:drawableBottom="@drawable/cons_ico"
    android:textSize="25sp"
    android:textStyle="bold"
    android:textAlignment="center"
    android:textColor="@color/colorPrimaryDark"

    />
</RelativeLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:background="@color/colorWhite"
    android:orientation="horizontal"
    >
    <Button
        android:id="@+id/BtnHist"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginEnd="100dp"
        android:layout_marginStart="100dp"
        android:layout_marginTop="10dp"
        android:drawableStart="@drawable/historial_ico"
        android:drawableEnd="@drawable/historial_ico"
        android:drawableTint="@color/colorWhite"
        android:text="HISTORIAL"
        android:textAlignment="center"
        android:textColor="@color/colorWhite"
        android:textStyle="bold"
        />

    </LinearLayout>

</LinearLayout>

</ScrollView>

</android.support.constraint.ConstraintLayout>

```

2.7.1.1.2. activity_historial_dia

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.HistorialDiaActivity">

<LinearLayout
    android:layout_width="match_parent"
    android:id="@+id/TopHistDay"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toTopOf="parent"
    android:orientation="vertical"
    >
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >
        <LinearLayout
            android:id="@+id/backgroundTOPSolar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:layout_constraintTop_toTopOf="parent"
            android:orientation="horizontal"
            android:background="@color/colorPrimaryDark"
            >
            <TextView
                android:id="@+id/TVFechaSolar"
                android:layout_width="match_parent"
                android:layout_height="30dp"
                android:layout_marginTop="10dp"
                android:layout_marginEnd="10dp"
                android:background="@color/colorPrimaryDark"
                android:textColor="@color/colorWhite"
                android:text="Historial"
                android:textSize="20sp"
                android:textAlignment="textEnd"
            />
        </LinearLayout>
        <Button
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:background="@drawable/historial_ico"
            android:layout_marginStart="10dp"
            android:layout_marginTop="10dp"
            android:backgroundTint="@color/colorWhite"
        />
    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="205dp"
        android:background="@color/colorPrimaryDark">
        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/LCPowerSolar"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:layout_marginTop="0dp"
            android:background="@color/colorPrimaryDark" />
    </RelativeLayout>
</LinearLayout>

<ScrollView

```

```

    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:id="@+id/ScrollViewSolar"
    android:background="@color/colorWhite"
    app:layout_constraintTop_toBottomOf="@+id/TopHistDay"
    app:layout_constraintBottom_toTopOf="@id/LayoutBottom"
  >

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
  >

    <RelativeLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content">

      <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/LCBatterySolar"
        android:layout_marginBottom="0dp"
        android:layout_marginTop="8dp"
        android:layout_width="match_parent"
        android:background="@color/Transparent"
        android:layout_height="180dp"/>

    </RelativeLayout>

    <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <TextView
        android:id="@+id/ProdActualSolar"
        android:text="0 Wh"
        android:layout_marginStart="25dp"
        android:layout_centerVertical="true"
        android:layout_height="wrap_content"
        android:layout_width="160dp"
        android:drawableBottom="@drawable/prod_ico"
        android:textSize="25sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"

      />

      <TextView
        android:id="@+id/ConsumoActualSolar"
        android:text="0 Wh"
        android:layout_marginEnd="25dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_height="wrap_content"
        android:layout_width="160dp"
        android:drawableBottom="@drawable/cons_ico"
        android:textSize="25sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"

      />
    </RelativeLayout>

  </LinearLayout>

</ScrollView>

<RelativeLayout
  android:id="@+id/LayoutBottom"
  android:layout_width="match_parent"
  android:layout_height="80dp"

```

```

        android:layout_gravity="center"
        app:layout_constraintBottom_toBottomOf="parent"
        android:background="@color/colorWhite"
    >

    <TextView
        android:id="@+id/FechaBottom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="06/08/2019"
        android:textColor="@color/colorPrimaryDark"
        android:gravity="center"
        android:textSize="25sp"
        android:layout_centerInParent="true"
    />

    <Button
        android:id="@+id/BtnFecha"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/Transparent"

    />

    <Button
        android:id="@+id/BtnAyer"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_centerVertical="true"
        android:background="@drawable/ic_back"
        android:backgroundTint="@color/colorPrimaryDark"
        android:layout_marginStart="15dp"

    />

    <Button
        android:id="@+id/BtnMañana"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:background="@drawable/ic_next"
        android:backgroundTint="@color/colorPrimaryDark"
        android:layout_marginEnd="15dp"
    />

</RelativeLayout>
</android.support.constraint.ConstraintLayout>

```


2.7.1.1.3. activity_riego

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.RiegoActivity">
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@color/colorPrimaryDark"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0">
<Button
    android:id="@+id/button_A_R"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:layout_weight="1"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_sun"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonA"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
    />
<Button
    android:id="@+id/button_B_R"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:layout_weight="1"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_riego"
    android:drawableTint="@color/colorBTN_ON"
    android:text="@string/ButtonB"
    android:textColor="@color/colorBTN_ON"
    android:background="@color/colorBTN_BG"
    android:textStyle="bold"
    />
<Button
    android:id="@+id/button_C_R"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:layout_weight="1"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_piscina"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonC"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
    />

```

```

<Button
    android:id="@+id/button_D_R"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:layout_weight="1"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_control3"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonD"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
/>
</LinearLayout>
<LinearLayout
    android:id="@+id/toplayoutriego"
    android:layout_width="match_parent"
    android:layout_height="130dp"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    android:paddingTop="0dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:background="@color/colorPrimaryDark"
    >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Control de riego"
            android:textSize="20sp"
            android:textColor="@color/colorWhite"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="10dp"
        />
        <ImageView
            android:id="@+id/imageView"
            android:layout_width="120dp"
            android:layout_height="match_parent"
            android:background="@drawable/ic_riego"
            android:layout_marginTop="10dp"
            android:layout_marginStart="20dp"
            android:layout_marginBottom="20dp"
        />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="A"
            android:textSize="25sp"
            android:textStyle="bold"
            android:textColor="@color/colorPrimaryDark"
            android:layout_alignParentStart="true"
            android:layout_marginStart="30dp"
            android:layout_marginTop="40dp"
        />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="B"
            android:textSize="25sp"
            android:textStyle="bold"
            android:textColor="@color/colorPrimaryDark"
            android:layout_alignParentStart="true"
            android:layout_marginStart="30dp"
            android:layout_marginTop="75dp"
        />
    </RelativeLayout>

```

```

</LinearLayout>
<LinearLayout
    android:id="@+id/LayouttextNext"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/toplayoutriego"
    android:background="@color/colorWhite"
    >
    <TextView
        android:id="@+id/TVCabecera"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="18sp"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"
        android:textStyle="bold"
        android:layout_margin="10dp"
        />

</LinearLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/LayouttextNext"
    app:layout_constraintVertical_bias="0.0"
    android:background="@color/colorWhite"
    >

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/ScrollviewSolar"
        android:background="@color/colorWhite"
        >

        <TextView
            android:id="@+id/TextInfoRiego"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:textSize="18sp"
            android:textStyle="normal"
            android:textColor="@color/colorPrimaryDark"
            android:text=""
            />

    </ScrollView>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:backgroundTint="@color/colorPrimaryDark"
        android:foregroundTint="@color/colorPrimary"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="15dp"
        app:borderWidth="0dp"
        android:src="@drawable/btn_riego"
        android:tint="@color/colorWhite"
        />

</RelativeLayout>
</android.support.constraint.ConstraintLayout>

```

2.7.1.1.4. activity_menuriego

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.MenuRiegoActivity">

<LinearLayout
    android:id="@+id/toplayoutpiscina"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    android:paddingTop="0dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimaryDark"
    >

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Menú de riego"
    android:textSize="20sp"
    android:textColor="@color/colorWhite"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"
    />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="200dp"
    android:layout_height="match_parent"
    android:layout_marginTop="30dp"
    android:background="@drawable/ic_riego"
    android:layout_marginStart="10dp"
    android:layout_marginBottom="10dp"
    android:backgroundTint="@color/colorPrimary"
    />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="A"
    android:textSize="25sp"
    android:textStyle="bold"
    android:textColor="@color/colorPrimaryDark"
    android:layout_alignParentStart="true"
    android:layout_marginStart="30dp"
    android:layout_marginTop="88dp"
    />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="B"
    android:textSize="25sp"
    android:textStyle="bold"
    android:textColor="@color/colorPrimaryDark"
    android:layout_alignParentStart="true"
    android:layout_marginStart="30dp"
    android:layout_marginTop="145dp"

```

```

        />

    </RelativeLayout>

</LinearLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/toplayoutpiscina"
    app:layout_constraintVertical_bias="0.0"
    android:background="@color/colorWhite"
    >

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/ScrollviewSolar"
        android:background="@color/colorWhite"
        >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            >
            <Button
                android:id="@+id/BtnCancel"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Borrar programas para hoy"
                android:textSize="16sp"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:layout_marginTop="8dp"
                android:layout_marginBottom="4dp"
                />

            <Button
                android:id="@+id/BtnBorrar"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Borrar programaciones"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:textSize="16sp"
                android:layout_marginTop="4dp"
                android:layout_marginBottom="4dp"
                />

            <Button
                android:id="@+id/BtnNuevo"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Nueva programación"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:textSize="16sp"
                android:layout_marginTop="4dp"
                android:layout_marginBottom="4dp"
                />

            <Button
                android:id="@+id/BtnRegar"
                android:layout_width="match_parent"
                android:layout_height="120dp"
                android:text="Regar ahora"
                android:background="@color/colorPrimary"

```

```
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
    />

    <Button
        android:id="@+id/Btnparar"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:text="Apagar Riego"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
    />

</LinearLayout>

</ScrollView>

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

2.7.1.1.5. activity_prog_riego

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Ativities.ProgRiegoActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorWhite"
    >

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:background="@color/colorPrimaryDark"
    >

    <Button
        android:id="@+id/btnCancel"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        android:background="@drawable/btn_cancel"
        android:backgroundTint="@color/colorWhite"
    />

    <Button
        android:id="@+id/btnOK"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"
        android:layout_alignParentEnd="true"
        android:background="@drawable/btn_ok"
        android:backgroundTint="@color/colorWhite"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nuevo programa&#xA;riego"
        android:textAlignment="center"
        android:textColor="@color/colorWhite"
        android:textSize="20sp"
        android:layout_centerInParent="true"
    />

</RelativeLayout>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="70dp"
>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
    >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
    >

```

```

android:layout_marginTop="10dp"
>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Duración"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="20sp"
    android:textAlignment="center"
/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >

    <NumberPicker
        android:id="@+id/np"
        android:layout_width="65dp"
        android:layout_height="180dp"
        android:theme="@style/NumberPickerStyle"
    />

    <TextView
        android:layout_width="30dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text=":"
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="30sp"
        android:textStyle="bold"
    />

    <NumberPicker
        android:id="@+id/n2"
        android:layout_width="65dp"
        android:layout_height="180dp"
        android:theme="@style/NumberPickerStyle"
    />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="20dp"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="65dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="h"
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
    />
    <TextView
        android:layout_width="30dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text=""
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
        android:textStyle="bold"
    />
    <TextView
        android:layout_width="65dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="min"
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"

```



```

        android:textSize="15sp"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:background="@color/colorWhite"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Lado A"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginEnd="10dp"
    />
    <ToggleButton
        android:id="@+id/SWRa"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_marginEnd="10dp"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Lado B"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginEnd="10dp"
        android:layout_marginStart="10dp"
    />
    <ToggleButton
        android:id="@+id/SWRB"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
    />
</LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/colorWhite"
    >
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:background="@color/colorWhite"
        android:layout_marginBottom="0.5dp"
        android:layout_marginTop="0.5dp"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hora"

```

```

        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        android:textStyle="bold"
    />

    <TextView
        android:id="@+id/Hora"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="00:00"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="22sp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="22dp"
    />

    <Button
        android:id="@+id/BtnHora"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/Transparent"
    />

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    android:layout_marginBottom="0.5dp"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Se repite"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        android:textStyle="bold"
    />

    <ToggleButton
        android:id="@+id/SWRepe"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RelativeLayout
        android:id="@+id/LayoutUnico"
        android:layout_width="match_parent"
        android:layout_height="50dp"

```

```

        android:layout_gravity="center"
        android:background="@color/colorWhite"
        android:layout_marginBottom="0.5dp"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fecha"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        android:textStyle="bold"
    />

    <TextView
        android:id="@+id/Fecha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hoy"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="22sp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="22dp"
    />

    <Button
        android:id="@+id/BtnFecha"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/Transparent"
    />

</RelativeLayout>

<LinearLayout
    android:id="@+id/LayoutRepetitivo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/colorWhite"
    >
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:background="@color/colorWhite"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Lunes"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="18sp"
            android:layout_alignParentStart="true"
            android:layout_centerVertical="true"
            android:layout_marginStart="20dp"
        />

        <ToggleButton
            android:id="@+id/SWLunes"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:background="@drawable/myswitch1"
            android:text=""
            android:textOff=""
            android:textOn=""
            android:layout_alignParentEnd="true"
            android:layout_marginEnd="20dp"
        />
    </RelativeLayout>
</LinearLayout>

```

```

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Martes"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        />

    <ToggleButton
        android:id="@+id/SWMartes"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
        />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Miércoles"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        />

    <ToggleButton
        android:id="@+id/SWmiercoles"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
        />

</RelativeLayout>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jueves"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        />

    <ToggleButton
        android:id="@+id/SWJueves"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
        />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Viernes"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
        />

    <ToggleButton
        android:id="@+id/SWViernes"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
        />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sábado"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"

```

```

        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />
    <ToggleButton
        android:id="@+id/SWSabado"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Domingo"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

    <ToggleButton
        android:id="@+id/SWDomingo"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
</LinearLayout>

</RelativeLayout>
</LinearLayout>
</LinearLayout>
</ScrollView>
</RelativeLayout>
</android.support.constraint.ConstraintLayout>

```

2.7.1.1.6. activity_piscina

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.PiscinaActivity">

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@color/colorPrimaryDark"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0">

<Button
    android:id="@+id/button_A_P"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:layout_weight="1"
    android:background="@color/colorBTN_BG"
    android:drawableTop="@drawable/btn_sun"
    android:drawablePadding="0dp"
    android:drawableTint="@color/colorBTN_OFF"
    android:gravity="top|center_horizontal"
    android:paddingTop="10dp"
    android:text="@string/ButtonA"
    android:textAllCaps="false"
    android:textColor="@color/colorBTN_OFF"
    android:textSize="12sp" />

<Button
    android:id="@+id/button_B_P"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:layout_weight="1"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_riego"
    android:drawableTint="@color/colorBTN_OFF"
    android:text="@string/ButtonB"
    android:textColor="@color/colorBTN_OFF"
    android:background="@color/colorBTN_BG"
    android:textStyle="bold"
/>

<Button
    android:id="@+id/button_C_P"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="0dp"
    android:paddingTop="10dp"
    android:gravity="top|center_horizontal"
    android:drawablePadding="0dp"
    android:layout_weight="1"
    android:textAllCaps="false"
    android:textSize="12sp"
    android:drawableTop="@drawable/btn_piscina"
    android:drawableTint="@color/colorBTN_ON"

```

```

        android:text="@string/ButtonC"
        android:textColor="@color/colorBTN_ON"
        android:background="@color/colorBTN_BG"
    />

    <Button
        android:id="@+id/button_D_P"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:layout_weight="1"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_control3"
        android:drawableTint="@color/colorBTN_OFF"
        android:text="@string/ButtonD"
        android:textColor="@color/colorBTN_OFF"
        android:background="@color/colorBTN_BG"
    />
</LinearLayout>

<LinearLayout
    android:id="@+id/toplayoutpiscina"
    android:layout_width="match_parent"
    android:layout_height="130dp"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    android:paddingTop="0dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:background="@color/colorPrimaryDark"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Control de depuradora"
            android:textSize="20sp"
            android:textColor="@color/colorWhite"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="10dp"
        />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="145dp"
            android:layout_height="match_parent"
            android:layout_marginBottom="10dp"
            android:layout_marginTop="10dp"
            android:background="@drawable/ic_swpool"
            android:backgroundTint="@color/colorPrimary"
        />

    </RelativeLayout>

</LinearLayout>

<LinearLayout
    android:id="@+id/LayouttextNext"
    android:orientation="vertical"
    android:layout_width="match_parent"

```



```

        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/toplayoutpiscina"
        android:background="@color/colorWhite"
    >
    <TextView
        android:id="@+id/TVCabecera"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="18sp"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"
        android:textStyle="bold"
        android:layout_margin="10dp"
    />

</LinearLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintBottom_toTopOf="@+id/linearlayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/LayouttextNext"
    app:layout_constraintVertical_bias="0.0"
    android:background="@color/colorWhite"
>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/ScrollviewSolar"
        android:background="@color/colorWhite"
    >

        <TextView
            android:id="@+id/TextDep"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:textSize="18sp"
            android:textStyle="normal"
            android:textColor="@color/colorPrimaryDark"
            android:text=""
        />

    </ScrollView>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:backgroundTint="@color/colorPrimaryDark"
        android:foregroundTint="@color/colorPrimary"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="15dp"
        app:borderWidth="0dp"
        android:src="@drawable/btn_piscina"
        android:tint="@color/colorWhite"
    />

</RelativeLayout>

</android.support.constraint.ConstraintLayout>

```

2.7.1.1.7. activity_menudep

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.MenuDepActivity">

<LinearLayout
    android:id="@+id/toplayoutpiscina"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    android:paddingTop="0dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorPrimaryDark"
        >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Menú de depuradora"
            android:textSize="20sp"
            android:textColor="@color/colorWhite"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="10dp"
            />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="220dp"
            android:layout_height="match_parent"
            android:layout_marginTop="40dp"
            android:background="@drawable/ic_swpool"
            android:backgroundTint="@color/colorPrimary"
            />

    </RelativeLayout>

</LinearLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/toplayoutpiscina"
    app:layout_constraintVertical_bias="0.0"
    android:background="@color/colorWhite"
    >

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"

```

```

android:id="@+id/ScrollViewSolar"
android:background="@color/colorWhite"
>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >
    <Button
        android:id="@+id/BtnCancel"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Borrar programas para hoy"
        android:textSize="16sp"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="4dp"
        />

    <Button
        android:id="@+id/BtnBorrar"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Borrar programaciones"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="16sp"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        />

    <Button
        android:id="@+id/BtnNuevo"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Nueva programación"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="16sp"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        />

    <Button
        android:id="@+id/BtnDepurar"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:text="depurar ahora"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        />

    <Button
        android:id="@+id/Btnparar"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:text="Apagar depuradora"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        />

    </LinearLayout>
</ScrollView>
</RelativeLayout>
</android.support.constraint.ConstraintLayout>

```

2.7.1.1.8. activity_prog_dep

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Ativities.ProgDepActivity">

<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/colorWhite"
>

<RelativeLayout
android:layout_width="match_parent"
android:layout_height="70dp"
android:background="@color/colorPrimaryDark"
>

<Button
android:id="@+id/btnCancel"
android:layout_width="30dp"
android:layout_height="30dp"
android:layout_centerVertical="true"
android:layout_marginStart="20dp"
android:background="@drawable/btn_cancel"
android:backgroundTint="@color/colorWhite"
/>

<Button
android:id="@+id/btnOK"
android:layout_width="30dp"
android:layout_height="30dp"
android:layout_centerVertical="true"
android:layout_marginEnd="20dp"
android:layout_alignParentEnd="true"
android:background="@drawable/btn_ok"
android:backgroundTint="@color/colorWhite"
/>

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Nuevo programa&#xA;depuradora"
android:textColor="@color/colorWhite"
android:textSize="20sp"
android:textAlignment="center"
android:layout_centerInParent="true"
/>

</RelativeLayout>

<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="70dp"
>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
android:gravity="center"
android:background="@color/colorWhite"

```

```

android:layout_marginTop="10dp"
>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Duración"
android:textColor="@color/colorPrimaryDark"
android:textSize="20sp"
android:textAlignment="center"
/>
<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal"
>

<NumberPicker
    android:id="@+id/np"
    android:layout_width="65dp"
    android:layout_height="180dp"
    android:theme="@style/NumberPickerStyle"
/>

<TextView
    android:layout_width="30dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text=":"
    android:textColor="@color/colorPrimaryDark"
    android:textAlignment="center"
    android:textSize="30sp"
    android:textStyle="bold"
/>

<NumberPicker
    android:id="@+id/n2"
    android:layout_width="65dp"
    android:layout_height="180dp"
    android:theme="@style/NumberPickerStyle"

/>

</LinearLayout>

<LinearLayout
android:layout_width="wrap_content"
android:layout_height="20dp"
android:orientation="horizontal"
>
<TextView
    android:layout_width="65dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="h"
    android:textColor="@color/colorPrimaryDark"
    android:textAlignment="center"
    android:textSize="15sp"
/>
<TextView
    android:layout_width="30dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text=""
    android:textColor="@color/colorPrimaryDark"
    android:textAlignment="center"
    android:textSize="15sp"
    android:textStyle="bold"
/>
<TextView
    android:layout_width="65dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="min"
    android:textColor="@color/colorPrimaryDark"
    android:textAlignment="center"

```

```

        android:textSize="15sp"
    />
</LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/colorWhite"
    >
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:background="@color/colorWhite"
        android:layout_marginBottom="0.5dp"
        android:layout_marginTop="0.5dp"
        >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hora"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="20sp"
            android:layout_alignParentStart="true"
            android:layout_centerVertical="true"
            android:layout_marginStart="20dp"
            android:textStyle="bold"
            />

        <TextView
            android:id="@+id/Hora"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="00:00"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="22sp"
            android:layout_alignParentEnd="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="22dp"
            />

        <Button
            android:id="@+id/BtnHora"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="@color/Transparent"
            />

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:background="@color/colorWhite"
        android:layout_marginBottom="0.5dp"
        >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Se repite"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="20sp"
            android:layout_alignParentStart="true"
            android:layout_centerVertical="true"
            android:layout_marginStart="20dp"
            android:textStyle="bold"
            />
    </RelativeLayout>
</LinearLayout>

```

```

<ToggleButton
    android:id="@+id/SWRepe"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/myswitch1"
    android:text=""
    android:textOff=""
    android:textOn=""
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="20dp"
/>

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

<RelativeLayout
    android:id="@+id/LayoutUnico"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
    android:layout_marginBottom="0.5dp"
>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fecha"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="20sp"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:layout_marginStart="20dp"
    android:textStyle="bold"
/>

<TextView
    android:id="@+id/Fecha"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hoy"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="22sp"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:layout_marginEnd="22dp"
/>

<Button
    android:id="@+id/BtnFecha"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/Transparent"
/>

</RelativeLayout>

<LinearLayout
    android:id="@+id/LayoutRepetitivo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/colorWhite"
>
<RelativeLayout
    android:layout_width="match_parent"

```

```

        android:layout_height="50dp"
        android:layout_gravity="center"
        android:background="@color/colorWhite"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lunes"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

    <ToggleButton
        android:id="@+id/SWLunes"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Martes"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

    <ToggleButton
        android:id="@+id/SWMartes"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

```



```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Miércoles"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="18sp"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:layout_marginStart="20dp"
/>

<ToggleButton
    android:id="@+id/SWmiercoles"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/myswitch1"
    android:text=""
    android:textOff=""
    android:textOn=""
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="20dp"
/>

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jueves"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="18sp"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:layout_marginStart="20dp"
/>

<ToggleButton
    android:id="@+id/SWJueves"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/myswitch1"
    android:text=""
    android:textOff=""
    android:textOn=""
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="20dp"
/>

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Viernes"
    android:textColor="@color/colorPrimaryDark"

```

```

        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

    <ToggleButton
        android:id="@+id/SWViernes"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sábado"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

    <ToggleButton
        android:id="@+id/SWSabado"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp"
    />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:background="@color/colorWhite"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Domingo"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:layout_marginStart="20dp"
    />

```

```
<ToggleButton
    android:id="@+id/SWDomingo"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@drawable/myswitch1"
    android:text=""
    android:textOff=""
    android:textOn=""
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="20dp"
/>

    </RelativeLayout>
</LinearLayout>
</RelativeLayout>

</LinearLayout>
</LinearLayout>
</ScrollView>

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

2.7.1.1.9. activity_otros

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activities.OtrosActivity">

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@color/colorPrimaryDark"
    android:orientation="horizontal"
    android:layout_alignParentBottom="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0">

    <Button
        android:id="@+id/button_A_0"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:layout_weight="1"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_sun"
        android:drawableTint="@color/colorBTN_OFF"
        android:text="@string/ButtonA"
        android:textColor="@color/colorBTN_OFF"
        android:background="@color/colorBTN_BG"
    />

    <Button
        android:id="@+id/button_B_0"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:layout_weight="1"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_riego"
        android:drawableTint="@color/colorBTN_OFF"
        android:text="@string/ButtonB"
        android:textColor="@color/colorBTN_OFF"
        android:background="@color/colorBTN_BG"
    />

    <Button
        android:id="@+id/button_C_0"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:layout_weight="1"
        android:textAllCaps="false"
        android:textSize="12sp"

```

```

        android:drawableTop="@drawable/btn_piscina"
        android:drawableTint="@color/colorBTN_OFF"
        android:text="@string/ButtonC"
        android:textColor="@color/colorBTN_OFF"
        android:background="@color/colorBTN_BG"
    />

    <Button
        android:id="@+id/button_D_O"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="0dp"
        android:paddingTop="10dp"
        android:gravity="top|center_horizontal"
        android:drawablePadding="0dp"
        android:layout_weight="1"
        android:textAllCaps="false"
        android:textSize="12sp"
        android:drawableTop="@drawable/btn_control13"
        android:drawableTint="@color/colorBTN_ON"
        android:text="@string/ButtonD"
        android:textColor="@color/colorBTN_ON"
        android:background="@color/colorBTN_BG"
        android:textStyle="bold"
    />
</LinearLayout>

<LinearLayout
    android:id="@+id/toplayoutriego"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:orientation="vertical"
    android:paddingTop="0dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    android:background="@color/colorPrimary"
>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:background="@color/colorPrimaryDark"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Otros controles"
            android:textSize="20sp"
            android:textColor="@color/colorWhite"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="10dp"
        />

    </RelativeLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Estado de los controles:"
        android:textStyle="bold"
        android:layout_margin="5dp"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
    />

    <TextView
        android:id="@+id/TextNivel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:text="- Nivel depósito: Medio"
        android:layout_margin="5dp"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
    />

    <TextView
        android:id="@+id/TextStatusDeposito"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="- Estado depósito: No se está llenando"
        android:layout_margin="5dp"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
    />

    <TextView
        android:id="@+id/TextPresion"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="- Estado presión: Encendida"
        android:layout_margin="5dp"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
    />

</LinearLayout>

<RelativeLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:id="@+id/linearLayoutRiego"
    android:background="@color/colorWhite"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/toplayoutriego"
    app:layout_constraintVertical_bias="0.0">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/ScrollViewSolar"
        android:background="@color/colorWhite"
    >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginTop="10dp"
        >

            <Button
                android:id="@+id/BtnLuces"
                android:layout_width="match_parent"
                android:layout_height="120dp"
                android:text="Control de luces (próximamente)"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:textSize="20sp"
                android:textStyle="bold"
                android:layout_marginTop="4dp"
                android:layout_marginBottom="4dp"
            />

            <Button
                android:id="@+id/BtnAgua"
                android:layout_width="match_parent"
                android:layout_height="120dp"
                android:text="Control de agua"

```

```
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
    />

    <Button
        android:id="@+id/BtnAjustes"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:text="Ajustes Automáticos"
        android:background="@color/colorPrimary"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
    />

    </LinearLayout>
</ScrollView>

</RelativeLayout>

</android.support.constraint.ConstraintLayout>
```

2.7.1.1.10. dialog_ajustes_auto

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@color/colorPrimaryDark"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ajustes automáticos"
        android:textColor="@color/colorWhite"
        android:textSize="18sp"
        android:layout_marginTop="10dp"
        android:textAlignment="center"
        android:layout_marginBottom="10dp"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        >

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:layout_marginStart="20dp"
                android:text="Llenado depósito auto."
                android:textSize="18sp"
                android:textColor="@color/colorPrimaryDark"
                android:layout_marginEnd="80dp"
                />

            <ToggleButton
                android:id="@+id/SWLLA"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:background="@drawable/myswitch1"
                android:text=""
                android:textOff=""
                android:textOn=""
                android:layout_marginEnd="20dp"
                android:layout_alignParentEnd="true"
                />

            </RelativeLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:layout_marginStart="20dp"
                android:text="Presión automática"
                android:textSize="18sp"
                android:textColor="@color/colorPrimaryDark"
                android:layout_marginEnd="80dp"

```



```

        />

        <ToggleButton
            android:id="@+id/SWPR"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:background="@drawable/myswitch1"
            android:text=""
            android:textOff=""
            android:textOn=""
            android:layout_marginEnd="20dp"
            android:layout_alignParentEnd="true"
        />

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_marginStart="20dp"
            android:text="Gestión de excedentes (prox.)"
            android:textSize="18sp"
            android:textColor="@color/colorPrimaryDark"
            android:layout_marginEnd="80dp"
        />

        <ToggleButton
            android:id="@+id/SWEX"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:background="@drawable/myswitch1"
            android:text=""
            android:textOff=""
            android:textOn=""
            android:layout_marginEnd="20dp"
            android:layout_alignParentEnd="true"
        />

    </RelativeLayout>

</LinearLayout>

</LinearLayout>

```

2.7.1.1.11. dialog_calendar

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_gravity="center"
    android:orientation="vertical"
    android:background="@color/colorWhite"
    >

    <DatePicker
        android:id="@+id/DatePicker"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/TimePickerStyle"
        android:timePickerMode="clock"

        >

    </DatePicker>

</LinearLayout>
```

2.7.1.1.12. dialog_ctrl_agua

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@color/colorPrimaryDark"
    android:layout_gravity="center"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Seleccionar acción"
        android:textColor="@color/colorWhite"
        android:textSize="18sp"
        android:layout_marginTop="10dp"
        android:textAlignment="center"
        android:layout_marginBottom="10dp"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:background="@color/colorWhite"
        >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:gravity="center"
            android:background="@color/colorWhite"
            >

            <Button
                android:id="@+id/BtnLlenar"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:layout_weight="1"
                android:text="Llenar depósito"
                android:textSize="18sp"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:shadowRadius="0"
                android:stateListAnimator="@null"
                android:layout_margin="10dp"
                />

            <Button
                android:id="@+id/BtnPresion"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:layout_weight="1"
                android:text="Encender presión de agua"
                android:textSize="18sp"
                android:background="@color/colorPrimary"
                android:textColor="@color/colorPrimaryDark"
                android:shadowRadius="0"
                android:stateListAnimator="@null"
                android:layout_marginBottom="10dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                />

        </LinearLayout>
    </LinearLayout>
</LinearLayout>

```

2.7.1.1.13. dialog_duracion

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@color/colorPrimaryDark"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Duración"
        android:textColor="@color/colorWhite"
        android:textSize="18sp"
        android:layout_marginTop="10dp"
        android:textAlignment="center"
        android:layout_marginBottom="10dp"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        >

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            >

            <NumberPicker
                android:id="@+id/np"
                android:layout_width="65dp"
                android:layout_height="wrap_content"
                android:theme="@style/NumberPickerStyle"
                />

            <TextView
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text=";"
                android:textColor="@color/colorPrimaryDark"
                android:textAlignment="center"
                android:textSize="30sp"
                android:textStyle="bold"
                />

            <NumberPicker
                android:id="@+id/n2"
                android:layout_width="65dp"
                android:layout_height="wrap_content"

```

```

        android:theme="@style/NumberPickerStyle"

    />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="65dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="h"
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
    />
    <TextView
        android:layout_width="30dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text=""
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
        android:textStyle="bold"
    />
    <TextView
        android:layout_width="65dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="min"
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
    />

</LinearLayout>

</LinearLayout>

</LinearLayout>

```

2.7.1.1.14. dialog_reg_ahora

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@color/colorPrimaryDark"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Duración y lado"
        android:textColor="@color/colorWhite"
        android:textSize="18sp"
        android:layout_marginTop="10dp"
        android:textAlignment="center"
        android:layout_marginBottom="10dp"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        >
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            >
            <NumberPicker
                android:id="@+id/np"
                android:layout_width="65dp"
                android:layout_height="wrap_content"
                android:theme="@style/NumberPickerStyle"
                />
            <TextView
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text=":"
                android:textColor="@color/colorPrimaryDark"
                android:textAlignment="center"
                android:textSize="30sp"
                android:textStyle="bold"
                />
            <NumberPicker
                android:id="@+id/n2"
                android:layout_width="65dp"
                android:layout_height="wrap_content"
                android:theme="@style/NumberPickerStyle"
                />
        </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <TextView
            android:layout_width="65dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="h"
            android:textColor="@color/colorPrimaryDark"
            android:textAlignment="center"
            android:textSize="15sp"
            />
    <TextView

```

```

        android:layout_width="30dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text=""
        android:textColor="@color/colorPrimaryDark"
        android:textAlignment="center"
        android:textSize="15sp"
        android:textStyle="bold"
    />
<TextView
    android:layout_width="65dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="min"
    android:textColor="@color/colorPrimaryDark"
    android:textAlignment="center"
    android:textSize="15sp"
/>

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:background="@color/colorWhite"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Lado A"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginEnd="10dp"
    />
    <ToggleButton
        android:id="@+id/SWRa"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
        android:layout_marginEnd="10dp"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Lado B"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginEnd="10dp"
        android:layout_marginStart="10dp"
    />
    <ToggleButton
        android:id="@+id/SWRB"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/myswitch1"
        android:text=""
        android:textOff=""
        android:textOn=""
    />
</LinearLayout>

</LinearLayout>

</LinearLayout>

```

2.7.1.1.15. dialog_time

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@color/colorPrimaryDark"
    >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        >

        <TimePicker
            android:id="@+id/TimePicker"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/TimePickerStyle"
            android:timePickerMode="clock"

            >

        </TimePicker>

    </LinearLayout>
</LinearLayout>
```


2.7.1.2. Código Java

2.7.1.2.1. Activities

2.7.1.2.1.1. SolarActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.example.alexn.myapplication.ChartModels.LineChartBaterly;
import com.example.alexn.myapplication.ChartModels.LineChartPower;
import com.example.alexn.myapplication.FnModels.CuentaAtras;
import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RaspPiService;
import com.example.alexn.myapplication.ServerModels.RasPiInversor;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;

import java.text.DecimalFormat;
import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class SolarActivity extends AppCompatActivity {

    // _____ Declaración de variables del MAIN ACTIVITY
    Button btnRiego, btnPiscina, btnOtros; // A: Solar, B: Riego, C: Piscina, D:
    Otros
    LineChart LCPower, LCBaterly;
    LineChartPower LCP;
    LineChartBaterly LCB;
    int Year, Month, Day, LastID, progressP, progressC;
    RasPiInversor Data;
    ProgressBar PBCons, PBProd;

    ArrayList<Entry> Cons, Prod, Bat;
    Boolean Conexion;
    TextView ProdAct, ConsAct, PrInst, CoInst, BatAct;
    String Produccion, Consumo, P_Inst, C_Inst, Bateria;

    final HTTPAsincrono IniciaHTTP = new HTTPAsincrono();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_solar);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        IniciVariables();
        HTTPInversor("False");
        IniciaHTTP.execute();
        btnRiego.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CuentaAtras.Pulsaciones(true);
            }
        });
    }

```

```

        Intent intent = new Intent(SolarActivity.this, RiegoActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});
btnPiscina.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(SolarActivity.this, PiscinaActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});
btnOtros.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(SolarActivity.this, OtrosActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});

findViewById(R.id.BtnHist).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(SolarActivity.this,
HistorialDiaActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});

}

@Override
//
//
MÉTODO CUANTA PULSACIÓN BOTÓN ATRÁS
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        return super.onKeyDown(keyCode, event);
    }

    return super.onKeyDown(keyCode, event);
}

private void IniciVariables(){
    LastID = 0;
    Year = FormatoFecha.GetYear();
    Month = FormatoFecha.GetMonth();
    Day = FormatoFecha.GetMonthDay();
    btnRiego = findViewById(R.id.button_B_Solar);
    btnPiscina = findViewById(R.id.button_C_Solar);
    btnOtros = findViewById(R.id.button_D_Solar);
    LCBatery = findViewById(R.id.LCBaterySolar);
    LCPower = findViewById(R.id.LCPowerSolar);
    LCP = new LineChartPower(LCPower, Cons, Prod, SolarActivity.this);
    LCB = new LineChartBatery(LCBatery, Bat, SolarActivity.this);
    LCP.NoData();
    LCB.NoData();
    ProdAct = findViewById(R.id.ProdActualSolar);
    PrInst = findViewById(R.id.ProdInst);
    ConsAct = findViewById(R.id.ConsumoActualSolar);
    CoInst = findViewById(R.id.ConsumoInst);
}

```

```

        BatAct = findViewById(R.id.BatActualSolar);
        TextView Fecha = findViewById(R.id.TVFechaSolar);
        Fecha.setText("Hoy, " + FormatoFecha.getCurrentDate());
        PBCons = findViewById(R.id.barCons);
        PBProd = findViewById(R.id.bar1);
    }

    private void HTTPInversor(String Last){
        String D = Integer.toString(Day);
        String M = Integer.toString(Month);
        String Y = Integer.toString(Year);
        RaspPiService service = RasPi.getRasPi().create(RaspPiService.class);
        Call<RasPiInversor> RasPiCall = service.LeeInversorFecha(RasPi.RasPiKey, D, M,
Y, Last, Integer.toString(LastID));
        RasPiCall.enqueue(new Callback<RasPiInversor>() {
            @Override
            public void onResponse(Call<RasPiInversor> call, Response<RasPiInversor>
response) {
                Conexion = true;
                Data = response.body();
                if(Data.getPos() != null){
                    if (Data.getLastID() != 0){
                        Cons = LCP.NewArrayList(Data.getPos(),Data.getWC());
                        Prod = LCP.NewArrayList(Data.getPos(),Data.getWP());
                        Bat = LCB.NewArrayList(Data.getPos(),Data.getBat());
                        LCP.SetData2(Cons, Prod);
                        LCB.SetData2(Bat);
                        Produccion = PotenciaToStr(Data.getWPh(), true);
                        Consumo = PotenciaToStr(Data.getWCh(), true);
                        progressP = (int)Data.getWP()[Data.getWP().length-1];
                        progressC = (int)Data.getWC()[Data.getWC().length-1];
                        P_Inst = PotenciaToStr(Data.getWP()[Data.getWP().length-1],
false);
                        C_Inst = PotenciaToStr(Data.getWC()[Data.getWC().length-1],
false);
                        Bateria = BatToStr(Data.getBat()[Data.getBat().length-1]);
                        LastID = Data.getLastID();
                    }
                }else{
                    Bateria = "0%";
                    Produccion = "0 Wh";
                    Consumo = "0 Wh";
                    P_Inst = "0 W";
                    C_Inst = "0 W";
                    progressP = 5000;
                    progressC = 5000;
                    LCP.NoData();
                    LCB.NoData();
                }
                ProdAct.setText(Produccion);
                ConsAct.setText(Consumo);
                PrInst.setText(P_Inst);
                CoInst.setText(C_Inst);
                BatAct.setText(Bateria);
                PBCons.setProgress(progressC);
                PBProd.setProgress(progressP);
            }

            @Override
            public void onFailure(Call<RasPiInversor> call, Throwable t) {
                Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_LONG).show();
            }
        });
    }

    private String PotenciaToStr(float Power, boolean h){
        String Wh = "";
        if (h) Wh = "h";
        String Pow = "0 Wh";
        DecimalFormat formato = new DecimalFormat("#.##");
        if (Power >= 1000){
            float kP = Power/1000;
            Pow = formato.format(kP) + " kW" + Wh;
        }else{

```

```

        Pow = formato.format(Power) + "W" + Wh;
    }
    return Pow;
}

private String BatToStr(float Porcent){
    String Porcentaje = "0 Wh";
    DecimalFormat formato = new DecimalFormat("#.##");
    Porcentaje = formato.format(Porcent) + "%";
    return Porcentaje;
}

private class HTTPAsincrono extends AsyncTask<Void, Void, Void>{

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
        HTTPInversor("True");
        //Toast.makeText(SolarActivity.this, "Actualizado",
        Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onCancelled(Void aVoid) {
        super.onCancelled(aVoid);
    }

    @Override
    protected void onCancelled() {
        super.onCancelled();
        //Toast.makeText(SolarActivity.this, "Cancelado",
        Toast.LENGTH_SHORT).show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        int i = 0;
        while(!isCancelled() && i<900){
            publishProgress();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            i++;
        }
        return null;
    }
}
}

```

2.7.1.2.1.2. HistorialDiaActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.widget.DatePicker;
import android.widget.TextView;
import android.widget.Toast;

import com.example.alexn.myapplication.ChartModels.LineChartBaterly;
import com.example.alexn.myapplication.ChartModels.LineChartPower;
import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RaspPiService;
import com.example.alexn.myapplication.ServerModels.RasPiInversor;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Locale;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class HistorialDiaActivity extends AppCompatActivity {

    TextView FechaBottom;
    int Year, Month, Day;
    String FechaBott;
    RasPiInversor Data;
    LineChart LCPower, LCBaterly;
    LineChartPower LCP;
    LineChartBaterly LCB;
    ArrayList<Entry> Cons, Prod, Bat;
    Boolean EspResp;
    TextView ProdAct, ConsAct;
    String Produccion, Consumo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_historial_dia);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        FechaAyer();
        LCBaterly = findViewById(R.id.LCBaterlySolar);
        LCPower = findViewById(R.id.LCPowerSolar);
        EspResp = false;
        ProdAct = findViewById(R.id.ProdActualSolar);
        ConsAct = findViewById(R.id.ConsumoActualSolar);
        //HTTP
        CreateGraph();
        HTTPInversor();

        findViewById(R.id.BtnFecha).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DialogoFecha();
            }
        });
    }
}

```

```

findViewById(R.id.BtnAyer).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(!EspResp){
            RestalFecha();
            HTTPInversor();
        }
    }
});

findViewById(R.id.BtnMañana).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(!EspResp){
            if (AyerOMenor()) SumalFecha();
            if (!AyerOMenor()) FechaAyer();
            HTTPInversor();
        }
    }
});

}

private void DialogoFecha(){
    AlertDialog.Builder builder = new AlertDialog.Builder(HistorialDiaActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_calendar, null);

    final DatePicker DP = view.findViewById(R.id.DatePicker);

    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Year = DP.getYear();
            Month = DP.getMonth()+1;
            Day = DP.getDayOfMonth();
            if(AyerOMenor()){
                FechaBottom.setText(FormatoFecha.AnyDayFechaHist(Year, Month, Day));
            }
            else{
                FechaAyer();
                Toast.makeText(getApplicationContext(), "La fecha debe ser anterior
a hoy", Toast.LENGTH_LONG).show();
            }
            HTTPInversor();
        }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {

        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void RestalFecha(){
    Calendar cal = FormatoFecha.NewCalendar(Year, Month, Day);
    cal.add(Calendar.DATE, -1);
    Year = cal.get(Calendar.YEAR);
    Month = cal.get(Calendar.MONTH) + 1;
    Day = cal.get(Calendar.DAY_OF_MONTH);
    String DayStr = cal.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.getDefault());
    DayStr = DayStr.substring(0,1).toUpperCase() + DayStr.substring(1);
    FechaBott = DayStr + "\n" + Day + "/" + Month + "/" + Year;
    FechaBottom.setText(FechaBott);
}

private void FechaAyer(){

```

```

FechaBottom = findViewById(R.id.FechaBottom);
Year = FormatoFecha.GetYear();
Month = FormatoFecha.GetMonth();
Day = FormatoFecha.GetMonthDay();
RestalFecha();
}

private void SumalFecha(){
    Calendar cal = FormatoFecha.NewCalendar(Year, Month, Day);
    cal.add(Calendar.DATE, 1);
    Year = cal.get(Calendar.YEAR);
    Month = cal.get(Calendar.MONTH) + 1;
    Day = cal.get(Calendar.DAY_OF_MONTH);
    String DayStr = cal.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.getDefault());
    DayStr = DayStr.substring(0,1).toUpperCase() + DayStr.substring(1);
    FechaBott = DayStr + "\n" + Day + "/" + Month + "/" + Year;
    FechaBottom.setText(FechaBott);
}

private boolean AyerOMenor(){
    int YearAct = FormatoFecha.GetYear();
    int MonthAct = FormatoFecha.GetMonth();
    int DayAct = FormatoFecha.GetMonthDay();
    return Year < YearAct ||
        Month < MonthAct ||
        (Year == YearAct && Month == MonthAct && Day <= DayAct-1);
}

private void CreateGraph(){
    LCP = new LineChartPower(LCPower, Cons, Prod, HistorialDiaActivity.this);
    LCB = new LineChartBaterly(LCBaterly, Bat, HistorialDiaActivity.this);
    LCP.NoData();
    LCB.NoData();
}

private void HTTPInversor(){
    String D = Integer.toString(Day);
    String M = Integer.toString(Month);
    String Y = Integer.toString(Year);
    EspResp = true;
    RasPiService service = RasPi.getRasPi().create(RasPiService.class);
    Call<RasPiInversor> RasPiCall = service.LeeInversorFecha(RasPi.RasPiKey, D, M,
Y, "False", "");
    RasPiCall.enqueue(new Callback<RasPiInversor>() {
        @Override
        public void onResponse(Call<RasPiInversor> call, Response<RasPiInversor>
response) {
            EspResp = false;
            Data = response.body();
            if(Data.getPos() != null){
                Cons = LCP.NewArrayList(Data.getPos(),Data.getWC());
                Prod = LCP.NewArrayList(Data.getPos(),Data.getWP());
                Bat = LCB.NewArrayList(Data.getPos(),Data.getBat());
                LCP.SetData2(Cons, Prod);
                LCB.SetData2(Bat);
                Produccion = PotenciaHTToStr(Data.getWPh());
                Consumo = PotenciaHTToStr(Data.getWCh());
            }else{
                Produccion = "0 Wh";
                Consumo = "0 Wh";
                LCP.NoData();
                LCB.NoData();
            }
            ProdAct.setText(Produccion);
            ConsAct.setText(Consumo);
        }
    });
    @Override
    public void onFailure(Call<RasPiInversor> call, Throwable t) {
        EspResp = false;
        Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_LONG).show();
    }
}

```

```

    });
}

private String PotenciaHTToStr(float Power){
    String Pow = "0 Wh";
    DecimalFormat formato = new DecimalFormat("#.##");
    if (Power >= 1000){
        float kP = Power/1000;
        Pow = formato.format(kP) + " kWh";
    }else{
        Pow = formato.format(Power) + "Wh";
    }
    return Pow;
}

@Override
//
MÉTODO CUANTA PULSACIÓN BOTÓN ATRÁS
public boolean onKeyDown(int keyCode, KeyEvent event) {
    Intent intent = new Intent(HistorialDiaActivity.this, SolarActivity.class);
    startActivity(intent);
    overridePendingTransition(0,0);
    return super.onKeyDown(keyCode, event);
}
}

```


2.7.1.2.1.3. RiegoActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import com.example.alexn.myapplication.FnModels.CuentaAtras;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RasPiService;
import com.example.alexn.myapplication.ServerModels.RasPiData;
import com.example.alexn.myapplication.ServerModels.RasPiTask;
import com.example.alexn.myapplication.ServerModels.RasPiValores;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class RiegoActivity extends AppCompatActivity {
    private Button btnSolar, btnPiscina, btnOtros; // A: Solar, B: Riego, C:
    Piscina, D: Otros
    private FloatingActionButton MenuRiego;
    private SeekBar Barra1;
    private TextView Tareas, Cabecera;
    final HTTPAsincrono IniciaHTTP = new HTTPAsincrono();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_riego);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        btnSolar = findViewById(R.id.button_A_R);
        btnPiscina = findViewById(R.id.button_C_R);
        btnOtros = findViewById(R.id.button_D_R);
        MenuRiego = findViewById(R.id.fab);

        Tareas = findViewById(R.id.TextInfoRiego);
        Cabecera = findViewById(R.id.TVCabecera);

        HTTPViewTask();
        IniciaHTTP.execute();

        MenuRiego.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(RiegoActivity.this, MenuRiegoActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
                IniciaHTTP.cancel(true);
            }
        });

        btnSolar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CuentaAtras.Pulsaciones(true);
                Intent intent = new Intent(RiegoActivity.this, SolarActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

```

        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});
btnPiscina.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(RiegoActivity.this, PiscinaActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});
btnOtros.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(RiegoActivity.this, OtrosActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        IniciaHTTP.cancel(true);
    }
});
}

private void HTTPViewTask(){
    RaspPiService service = RaspPi.getRaspPi().create(RaspPiService.class);
    Call<RaspPiTask> RaspPiCall = service.ViewTask(RaspPi.RaspPiKey, "Riego");
    RaspPiCall.enqueue(new Callback<RaspPiTask>() {
        @Override
        public void onResponse(Call<RaspPiTask> call, Response<RaspPiTask> response) {
            RaspPiTask Riegos = response.body();
            Tareas.setText(Riegos.getTareas());
            Cabecera.setText(Riegos.getCabecera());
        }

        @Override
        public void onFailure(Call<RaspPiTask> call, Throwable t) {

        }
    });
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        // Esto es lo que hace mi botón al pulsar ir a atrás
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        return super.onKeyDown(keyCode, event);
    }
    return super.onKeyDown(keyCode, event);
}

private class HTTPAsincrono extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
        HTTPViewTask();
    }
}

```

```

        //Toast.makeText(SolarActivity.this, "Actualizado",
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onCancelled(Void aVoid) {
        super.onCancelled(aVoid);
    }

    @Override
    protected void onCancelled() {
        super.onCancelled();
        //Toast.makeText(SolarActivity.this, "Cancelado",
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        int i = 0;
        while(!isCancelled() && i<600){
            publishProgress();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            i++;
        }
        return null;
    }
}
}
}

```

2.7.1.2.1.4. MenuRiegoActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RasPiService;
import com.example.alexn.myapplication.ServerModels.RasPiTask;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MenuRiegoActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menuriego);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        findViewById(R.id.BtnCancel).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoBorrarDia();
            }
        });

        findViewById(R.id.BtnRegar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoAhora();
            }
        });

        findViewById(R.id.BtnNuevo).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MenuRiegoActivity.this,
                ProgRiegoActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        });

        findViewById(R.id.BtnBorrar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoBorrarTodo();
            }
        });

        findViewById(R.id.Btnparar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast toast = Toast.makeText(getApplicationContext(), "Enviando parada de
                riego...", Toast.LENGTH_SHORT);
                toast.show();
                //Código para parar riego
            }
        });
    }
}

```

```

        HTTPNewTask("Unico", "STR", FormatoFecha.GetMonthDay(),
FormatoFecha.GetMonth(), FormatoFecha.GetYear(), FormatoFecha.GetH(),
FormatoFecha.GetMin());
        Intent intent = new Intent(MenuRiegoActivity.this, RiegoActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
    });
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    Intent intent = new Intent(MenuRiegoActivity.this, RiegoActivity.class);
    startActivity(intent);
    overridePendingTransition(0,0);
    return super.onKeyDown(keyCode, event);
}

private void CuadroDialogoBorrarDia() {
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuRiegoActivity.this,
R.style.DialogBasicCustomStyle);
    builder.setTitle("Borrar programas");
    builder.setMessage("Eliminará todos los riegos automáticos previstos para
hoy.\n\n¿Continuar?")
        .setPositiveButton("Sí", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast toast = Toast.makeText(getApplicationContext(),"Borrando
programas...", Toast.LENGTH_SHORT);
                toast.show();
                //Código para borrar programas
                HTTPClear("Today");
                Intent intent = new Intent(MenuRiegoActivity.this,
RiegoActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
            }
        })
        .setCancelable(false).show();
}

private void CuadroDialogoBorrarTodo() {
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuRiegoActivity.this,
R.style.DialogBasicCustomStyle);
    builder.setTitle("Borrar programas");
    builder.setMessage("Eliminará todos los riegos automáticos
programados.\n\n¿Continuar?")
        .setPositiveButton("Sí", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast toast = Toast.makeText(getApplicationContext(),"Borrando
programas...", Toast.LENGTH_SHORT);
                toast.show();
                //Código para borrar programas
                HTTPClear("All");
                Intent intent = new Intent(MenuRiegoActivity.this,
RiegoActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
            }
        })
        .setCancelable(false).show();
}
}

```

```

private void CuadroDialogoAhora(){
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuRiegoActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    final View view = inflater.inflate(R.layout.dialog_reg_ahora, null);
    final NumberPicker np = view.findViewById(R.id.np);
    np.setMinValue(0);
    np.setMaxValue(4);
    np.setWrapSelectorWheel(true);

    final NumberPicker n2 = view.findViewById(R.id.n2);
    n2.setMinValue(0);
    n2.setMaxValue(59);
    n2.setWrapSelectorWheel(true);

    np.setOnValueChangedListener(new NumberPicker.OnValueChangeListener() {
        @Override
        public void onValueChange(NumberPicker numberPicker, int i, int il) {
            if(np.getValue() == 4) n2.setMaxValue(15);
            else n2.setMaxValue(59);
        }
    });

    final ToggleButton SWA = view.findViewById(R.id.SWRa);
    final ToggleButton SWB = view.findViewById(R.id.SWRB);
    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            int Tiempo = np.getValue()*60 + n2.getValue();

            String Comando;
            if(Tiempo>0){
                if (SWA.isChecked() || SWB.isChecked()){
                    Toast.makeText(getApplicationContext(),"Enviando orden...",
Toast.LENGTH_SHORT).show();
                    if (SWA.isChecked() && SWB.isChecked()) Comando = "RAB"+Tiempo;
                    else if (SWA.isChecked()) Comando = "RRA"+Tiempo;
                    else Comando = "RRB"+Tiempo;
                    HTTPNewTask("Unico", Comando, FormatoFecha.GetMonthDay(),
FormatoFecha.GetMonth(), FormatoFecha.GetYear(), FormatoFecha.GetH(),
FormatoFecha.GetMin());
                    Intent intent = new Intent(MenuRiegoActivity.this,
RiegoActivity.class);
                    startActivity(intent);
                    overridePendingTransition(0,0);
                }else{
                    Toast.makeText(getApplicationContext(),"No se ha seleccionado
ningún lado", Toast.LENGTH_SHORT).show();
                }
            }else{
                Toast.makeText(getApplicationContext(),"No se ha seleccionado el
tiempo", Toast.LENGTH_SHORT).show();
            }
        }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void HTTPClear(String Delete){
    RasPiService service = RasPi.getRasPi().create(RasPiService.class);
    Call<Void> RasPiCall = service.ClearTaskt(RasPi.RasPiKey, "Riego", Delete);
    RasPiCall.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {

```

```

        Toast.makeText(getApplicationContext(), "Borrado",
Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onFailure(Call<Void> call, Throwable t) {
        Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_SHORT).show();
    }
});

private void HTTPNewTask(String Tipo, String Comando, int Dia, int Mes, int Ano, int
Hora, int Min){

    RaspPiService service = RasPi.getRasPi().create(RaspPiService.class);
    Call<Void> RasPiCall = service.NewTaskt(RasPi.RasPiKey,
Tipo,Comando,Integer.toString(Dia),Integer.toString(Mes),Integer.toString(Ano),Integer.t
oString(Hora),Integer.toString(Min));
    RasPiCall.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            Toast.makeText(getApplicationContext(), "Enviado",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_SHORT).show();
        }
    });
}

}

```

2.7.1.2.1.5. ProgRiegoActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.DatePicker;
import android.widget.LinearLayout;
import android.widget.NumberPicker;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RasPiService;

import java.util.Date;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class ProgRiegoActivity extends AppCompatActivity {

    NumberPicker NP_H, NP_Min;
    TextView HoraInicio, FechaInicio;
    RelativeLayout LayoutUnico;
    LinearLayout LayoutRepe;
    ToggleButton SWUnicoRep, SWL, SWM, SWX, SWJ, SWV, SWS, SWD, SWA, SWB;
    int Year, Month, Day, H, Min, Tiempo;
    String Tipo, Comando;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prog_riego);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        IniciaVariables();
        ModoUnico();
        SetNumberPickers();

        NP_H.setOnValueChangedListener(new NumberPicker.OnValueChangeListener() {
            @Override
            public void onValueChange(NumberPicker numberPicker, int i, int il) {
                if (NP_H.getValue() == 4) NP_Min.setMaxValue(15);
                else NP_Min.setMaxValue(59);
            }
        });

        findViewById(R.id.BtnHora).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DialogoHora();
            }
        });

        findViewById(R.id.BtnFecha).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```



```

        DialogoFecha();
    }
});

SWUnicoRep.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (SWUnicoRep.isChecked()) ModoRepetitivo();
        else ModoUnico();
    }
});

findViewById(R.id.btnCancel).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(ProgRiegoActivity.this,
MenuRiegoActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
});

findViewById(R.id.btnOK).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //GUARDAR PROGRAMA Y MOSTRAR MENSAJE
        ProgramaOK();
    }
});
}

private void DialogoHora(){
    AlertDialog.Builder builder = new AlertDialog.Builder(ProgRiegoActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_time, null);

    final TimePicker TP = view.findViewById(R.id.TimePicker);
    TP.setIs24HourView(true);

    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            String HoraMin;
            H = TP.getHour();
            if (H < 10) HoraMin = "0" + H;
            else HoraMin = "" + H;
            Min = TP.getMinute();
            if (Min < 10) HoraMin = HoraMin + ":0" + Min;
            else HoraMin = HoraMin + ":" + Min;
            HoraInicio.setText(HoraMin);
        }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {

        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void DialogoFecha(){
    AlertDialog.Builder builder = new AlertDialog.Builder(ProgRiegoActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_calendar, null);

    final DatePicker DP = view.findViewById(R.id.DatePicker);

    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {

```

```

        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Year = DP.getYear();
            Month = DP.getMonth()+1;
            Day = DP.getDayOfMonth();
            FechaInicio.setText(FormatoFecha.AnyDayFecha(Year,Month,Day));
        }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {

        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void SetNumberPickers(){
    NP_H.setMinValue(0);
    NP_H.setMaxValue(4);
    NP_H.setWrapSelectorWheel(true);
    NP_Min.setMinValue(0);
    NP_Min.setMaxValue(59);
    NP_Min.setWrapSelectorWheel(true);
}

private void ModoUnico(){
    LayoutRepe.setVisibility(View.INVISIBLE);
    LayoutUnico.setVisibility(View.VISIBLE);
    LayoutRepe.getLayoutParams().height=0;
}

private void ModoRepetitivo(){
    LayoutRepe.setVisibility(View.VISIBLE);
    LayoutUnico.setVisibility(View.INVISIBLE);
    ViewGroup.LayoutParams params = LayoutRepe.getLayoutParams();
    params.height = ViewGroup.LayoutParams.WRAP_CONTENT;
    LayoutRepe.setLayoutParams(params);
}

private void IniciaVariables(){
    SWUnicoRep = findViewById(R.id.SWRepe);
    SWL = findViewById(R.id.SWLunes);
    SWM = findViewById(R.id.SWMartes);
    SWX = findViewById(R.id.SWMiercoles);
    SWJ = findViewById(R.id.SWJueves);
    SWV = findViewById(R.id.SWViernes);
    SWS = findViewById(R.id.SWSabado);
    SWD = findViewById(R.id.SWDomingo);
    SWA = findViewById(R.id.SWRa);
    SWB = findViewById(R.id.SWRB);
    NP_H = findViewById(R.id.np);
    NP_Min = findViewById(R.id.n2);
    HoraInicio = findViewById(R.id.Hora);
    FechaInicio = findViewById(R.id.Fecha);
    LayoutUnico = findViewById(R.id.LayoutUnico);
    LayoutRepe = findViewById(R.id.LayoutRepetitivo);
    HoraInicio.setText(FormatoFecha.getHour());
    FechaInicio.setText(FormatoFecha.getDate());
    Year = FormatoFecha.GetYear();
    Month = FormatoFecha.GetMonth();
    Day = FormatoFecha.GetMonthDay();
    H = FormatoFecha.GetH();
    Min = FormatoFecha.GetMin();
}

private void HTTPNewTask(String Tipo, String Comando, int Dia, int Mes, int Ano, int
Hora, int Min){

    RaspPiService service = RasPi.getRasPi().create(RaspPiService.class);
    Call<Void> RasPiCall = service.NewTaskt(RasPi.RasPiKey,
Tipo,Comando,Integer.toString(Dia),Integer.toString(Mes),Integer.toString(Ano),Integer.t
oString(Hora),Integer.toString(Min));

```


2.7.1.2.1.6. PiscinaActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.ToggleButton;

import com.example.alexn.myapplication.FnModels.CuentaAtras;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RaspPiService;
import com.example.alexn.myapplication.ServerModels.RasPiTask;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class PiscinaActivity extends AppCompatActivity {
    private Button btnSolar, btnRiego, btnOtros, btnSend; // A: Solar, B: Riego,
C: Piscina, D: Otros
    private FloatingActionButton NewDep;
    private ToggleButton Switch1;
    private EditText Valor_1, Valor_2;
    private static boolean SW1;

    private TextView Tareas, Cabecera;
    final HTTPAsincrono IniciaHTTP = new HTTPAsincrono();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_piscina);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        btnSolar = findViewById(R.id.button_A_P);
        btnRiego = findViewById(R.id.button_B_P);
        btnOtros = findViewById(R.id.button_D_P);
        Tareas = findViewById(R.id.TextDep);
        Cabecera = findViewById(R.id.TVCabecera);

        NewDep = findViewById(R.id.fab);

        HTTPViewTask();

        NewDep.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(PiscinaActivity.this, MenuDepActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        });

        btnSolar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CuentaAtras.Pulsaciones(true);
                Intent intent = new Intent(PiscinaActivity.this, SolarActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        });
    }
}

```

```

});
btnRiego.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(PiscinaActivity.this, RiegoActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
});
btnOtros.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CuentaAtras.Pulsaciones(true);
        Intent intent = new Intent(PiscinaActivity.this, OtrosActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
});
});

}

private void HTTPViewTask() {
    RaspPiService service = RaspPi.getRaspPi().create(RaspPiService.class);
    Call<RasPiTask> RasPiCall = service.ViewTask(RaspPi.RasPiKey, "Depuradora");
    RasPiCall.enqueue(new Callback<RasPiTask>() {
        @Override
        public void onResponse(Call<RasPiTask> call, Response<RasPiTask> response) {
            RasPiTask Riegos = response.body();
            Tareas.setText(Riegos.getTareas());
            Cabecera.setText(Riegos.getCabecera());
        }

        @Override
        public void onFailure(Call<RasPiTask> call, Throwable t) {

        }
    });
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        // Esto es lo que hace mi botón al pulsar ir a atrás
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
    return super.onKeyDown(keyCode, event);
}

private class HTTPAsincrono extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
        HTTPViewTask();
        //Toast.makeText(SolarActivity.this, "Actualizado",
        Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onCancelled(Void aVoid) {
        super.onCancelled(aVoid);
    }
}

```

```
    }

    @Override
    protected void onCancelled() {
        super.onCancelled();
        //Toast.makeText(SolarActivity.this, "Cancelado",
        Toast.LENGTH_SHORT).show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        int i = 0;
        while(!isCancelled() && i<600){
            publishProgress();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            i++;
        }
        return null;
    }
}
```

2.7.1.2.1.7. MenuDepActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.NumberPicker;
import android.widget.TimePicker;
import android.widget.Toast;

import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RasPiService;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MenuDepActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menudep);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        findViewById(R.id.BtnCancelar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoBorrar();
            }
        });

        findViewById(R.id.BtnDepurar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoAhora();
            }
        });

        findViewById(R.id.BtnBorrar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CuadroDialogoBorrarTodo();
            }
        });

        findViewById(R.id.BtnNuevo).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MenuDepActivity.this, ProgDepActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        });

        findViewById(R.id.Btnparar).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast toast = Toast.makeText(getApplicationContext(), "Enviando parada de
depuradora...", Toast.LENGTH_SHORT);
                toast.show();
                HTTPNewTask("Unico", "STD", FormatoFecha.GetMonthDay(),
FormatoFecha.GetMonth(), FormatoFecha.GetYear(), FormatoFecha.GetH(),
FormatoFecha.GetMin());
            }
        });
    }
}

```

```

        //Código para parar riego
        Intent intent = new Intent(MenuDepActivity.this, PiscinaActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
}

private void CuadroDialogoBorrar(){
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuDepActivity.this,
R.style.DialogBasicCustomStyle);
    builder.setTitle("Borrar programas");
    builder.setMessage("Eliminará todas las depuraciones automáticas previstas para
hoy.\n\n¿Continuar?")
        .setPositiveButton("Sí", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast toast = Toast.makeText(getApplicationContext(),"Borrando
programas...", Toast.LENGTH_SHORT);
                toast.show();
                //Código para borrar programas
                HTTPClear("Today");
                Intent intent = new Intent(MenuDepActivity.this, PiscinaActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

            }
        })
        .setCancelable(false).show();
}

private void CuadroDialogoBorrarTodo(){
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuDepActivity.this,
R.style.DialogBasicCustomStyle);
    builder.setTitle("Borrar programas");
    builder.setMessage("Eliminará todas las depuraciones
programados.\n\n¿Continuar?")
        .setPositiveButton("Sí", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast toast = Toast.makeText(getApplicationContext(),"Borrando
programas...", Toast.LENGTH_SHORT);
                toast.show();
                //Código para borrar programas
                HTTPClear("All");
                Intent intent = new Intent(MenuDepActivity.this,
PiscinaActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
            }
        }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

            }
        })
        .setCancelable(false).show();
}

private void CuadroDialogoAhora(){
    AlertDialog.Builder builder = new AlertDialog.Builder(MenuDepActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_duracion, null);

    final NumberPicker np = view.findViewById(R.id.np);
    np.setMinValue(0);
    np.setMaxValue(4);
    np.setWrapSelectorWheel(true);
}

```



```

final NumberPicker n2 = view.findViewById(R.id.n2);
n2.setMinValue(0);
n2.setMaxValue(59);
n2.setWrapSelectorWheel(true);

np.setOnValueChangedListener(new NumberPicker.OnValueChangeListener() {
    @Override
    public void onValueChange(NumberPicker numberPicker, int i, int il) {
        if(np.getValue() == 4) n2.setMaxValue(15);
        else n2.setMaxValue(59);
    }
});

builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        int Tiempo = np.getValue()*60 + n2.getValue();
        if(Tiempo>0){
            Toast.makeText(getApplicationContext(),"Enviando orden...",
Toast.LENGTH_SHORT).show();
            String Comando = "DEP"+Tiempo;
            HTTPNewTask("Unico", Comando, FormatoFecha.GetMonthDay(),
FormatoFecha.GetMonth(), FormatoFecha.GetYear(), FormatoFecha.GetH(),
FormatoFecha.GetMin());
            Intent intent = new Intent(MenuDepActivity.this,
PiscinaActivity.class);
            startActivity(intent);
            overridePendingTransition(0,0);
        }else{
            Toast.makeText(getApplicationContext(),"No se ha seleccionado el
tiempo", Toast.LENGTH_SHORT).show();
        }
    }
}).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
    }
}).setCancelable(false);

AlertDialog dialog = builder.create();
dialog.show();

}

private void HTTPClear(String Delete){
    RasPiService service = RasPi.getRasPi().create(RasPiService.class);
    Call<Void> RasPiCall = service.ClearTaskt(RasPi.RasPiKey, "Depuradora", Delete);
    RasPiCall.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            Toast.makeText(getApplicationContext(),"Borrado",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Toast.makeText(getApplicationContext(),"Error de conexión",
Toast.LENGTH_SHORT).show();
        }
    });
}

private void HTTPNewTask(String Tipo, String Comando, int Dia, int Mes, int Ano, int
Hora, int Min){

    RasPiService service = RasPi.getRasPi().create(RasPiService.class);

```

```

        Call<Void> RasPiCall = service.NewTaskt(RasPi.RasPiKey,
        Tipo,Comando,Integer.toString(Dia),Integer.toString(Mes),Integer.toString(Ano),Integer.t
        oString(Hora),Integer.toString(Min));
        RasPiCall.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(Call<Void> call, Response<Void> response) {
                Toast.makeText(getApplicationContext(),"Enviado",
                Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(getApplicationContext(),"Error de conexión",
                Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        Intent intent = new Intent(MenuDepActivity.this, PiscinaActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
        return super.onKeyDown(keyCode, event);
    }
}

```

2.7.1.2.1.8. ProgDepActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.DatePicker;
import android.widget.LinearLayout;
import android.widget.NumberPicker;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import com.example.alexn.myapplication.FnModels.FormatoFecha;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RasPiService;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class ProgDepActivity extends AppCompatActivity {

    NumberPicker NP_H, NP_Min;
    TextView HoraInicio, FechaInicio;
    RelativeLayout LayoutUnico;
    LinearLayout LayoutRepe;
    ToggleButton SWUnicoRep, SWL, SWM, SWX, SWJ, SWV, SWS, SWD;
    int Year, Month, Day, H, Min, Tiempo;
    String Tipo, Comando;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prog_dep);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        IniciaVariables();
        ModoUnico();
        SetNumberPickers();

        NP_H.setOnValueChangedListener(new NumberPicker.OnValueChangeListener() {
            @Override
            public void onValueChange(NumberPicker numberPicker, int i, int il) {
                if (NP_H.getValue() == 4) NP_Min.setMaxValue(15);
                else NP_Min.setMaxValue(59);
            }
        });

        findViewById(R.id.BtnHora).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DialogoHora();
            }
        });

        findViewById(R.id.BtnFecha).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DialogoFecha();
            }
        });
    }
}

```

```

SWUnicoRep.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (SWUnicoRep.isChecked()) ModoRepetitivo();
        else ModoUnico();
    }
});

findViewById(R.id.btnCancel).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(ProgDepActivity.this, MenuDepActivity.class);
        startActivity(intent);
        overridePendingTransition(0,0);
    }
});

findViewById(R.id.btnOK).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //GUARDAR PROGRAMA Y MOSTRAR MENSAJE
        ProgramaOK();
    }
});
}

private void DialogoHora(){
    AlertDialog.Builder builder = new AlertDialog.Builder(ProgDepActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_time, null);

    final TimePicker TP = view.findViewById(R.id.TimePicker);
    TP.setIs24HourView(true);

    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            String HoraMin;
            H = TP.getHour();
            if (H < 10) HoraMin = "0" + H;
            else HoraMin = "" + H;
            Min = TP.getMinute();
            if (Min < 10) HoraMin = HoraMin + ":0" + Min;
            else HoraMin = HoraMin + ":" + Min;
            HoraInicio.setText(HoraMin);
        }
    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void DialogoFecha(){
    AlertDialog.Builder builder = new AlertDialog.Builder(ProgDepActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_calendar, null);

    final DatePicker DP = view.findViewById(R.id.DatePicker);

    builder.setView(view).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Year = DP.getYear();
            Month = DP.getMonth()+1;
            Day = DP.getDayOfMonth();
        }
    });
}

```

```

        FechaInicio.setText (FormatoFecha.AnyDayFecha (Year,Month,Day));
    }
}).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

    }
}).setCancelable(false);

AlertDialog dialog = builder.create();
dialog.show();
}

private void SetNumberPickers(){
    NP_H.setMinValue(0);
    NP_H.setMaxValue(4);
    NP_H.setWrapSelectorWheel(true);
    NP_Min.setMinValue(0);
    NP_Min.setMaxValue(59);
    NP_Min.setWrapSelectorWheel(true);
}

private void ModoUnico(){
    LayoutRepe.setVisibility(View.INVISIBLE);
    LayoutUnico.setVisibility(View.VISIBLE);
    LayoutRepe.getLayoutParams().height=0;
}

private void ModoRepetitivo(){
    LayoutRepe.setVisibility(View.VISIBLE);
    LayoutUnico.setVisibility(View.INVISIBLE);
    ViewGroup.LayoutParams params = LayoutRepe.getLayoutParams();
    params.height = ViewGroup.LayoutParams.WRAP_CONTENT;
    LayoutRepe.setLayoutParams(params);
}

private void IniciaVariables(){
    SWUnicoRep = findViewById(R.id.SWRepe);
    SWL = findViewById(R.id.SWLunes);
    SWM = findViewById(R.id.SWMartes);
    SWX = findViewById(R.id.SWmiercoles);
    SWJ = findViewById(R.id.SWJueves);
    SWV = findViewById(R.id.SWViernes);
    SWS = findViewById(R.id.SWSabado);
    SWD = findViewById(R.id.SWDomingo);
    NP_H = findViewById(R.id.np);
    NP_Min = findViewById(R.id.n2);
    HoraInicio = findViewById(R.id.Hora);
    FechaInicio = findViewById(R.id.Fecha);
    LayoutUnico = findViewById(R.id.LayoutUnico);
    LayoutRepe = findViewById(R.id.LayoutRepetitivo);

    HoraInicio.setText (FormatoFecha.getHour());
    FechaInicio.setText (FormatoFecha.getDate());
    Year = FormatoFecha.GetYear();
    Month = FormatoFecha.GetMonth();
    Day = FormatoFecha.GetMonthDay();
    H = FormatoFecha.GetH();
    Min = FormatoFecha.GetMin();
}

private void HTTPNewTask(String Tipo, String Comando, int Dia, int Mes, int Ano, int
Hora, int Min){

    RaspPiService service = RasPi.getRasPi().create(RaspPiService.class);
    Call<Void> RasPiCall = service.NewTaskt(RasPi.RasPiKey,
Tipo,Comando,Integer.toString(Dia),Integer.toString(Mes),Integer.toString(Ano),Integer.t
oString(Hora),Integer.toString(Min));
    RasPiCall.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            Toast.makeText(getApplicationContext(),"Enviado",
Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_SHORT).show();
        }
    });
}

private boolean Tiempo(){
    Tiempo = NP H.getValue()*60 + NP_Min.getValue();
    Comando = "DEP"+Tiempo;
    return Tiempo>0;
}

private boolean Tipo(){
    int T = 0;
    if (SWUnicoRep.isChecked()){
        if(SWL.isChecked()) T = T + 1;
        if(SWM.isChecked()) T = T + 2;
        if(SWX.isChecked()) T = T + 4;
        if(SWJ.isChecked()) T = T + 8;
        if(SWV.isChecked()) T = T + 16;
        if(SWS.isChecked()) T = T + 32;
        if(SWD.isChecked()) T = T + 64;
        Tipo = Integer.toString(T);
    }else{
        T = 1;
        Tipo = "Unico";
    }
    return T!=0;
}

private void ProgramaOK(){
    if (Tiempo()){
        if(Tipo){
            Toast.makeText(getApplicationContext(), "Enviando",
Toast.LENGTH_SHORT).show();
            HTTPNewTask(Tipo, Comando, Day, Month, Year, H, Min);
            Intent intent = new Intent(ProgDepActivity.this,
PiscinaActivity.class);
            startActivity(intent);
            overridePendingTransition(0,0);
        }else{
            Toast.makeText(getApplicationContext(), "No se ha seleccionado ningún
día", Toast.LENGTH_SHORT).show();
        }
    }else{
        Toast.makeText(getApplicationContext(), "No se ha seleccionado ninguna
duración", Toast.LENGTH_SHORT).show();
    }
}
}
}

```

2.7.1.2.1.9. OtrosActivity

```

package com.example.alexn.myapplication.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

import com.example.alexn.myapplication.ChartModels.LineChartPower;
import com.example.alexn.myapplication.FnModels.ColorMPAChart;
import com.example.alexn.myapplication.FnModels.CuentaAtras;
import com.example.alexn.myapplication.ChartModels.LineChartBatery;
import com.example.alexn.myapplication.R;
import com.example.alexn.myapplication.RasPiServer.RasPi;
import com.example.alexn.myapplication.RasPiServer.RasPiServices.RaspPiService;
import com.example.alexn.myapplication.ServerModels.RasPiCtrlStat;
import com.example.alexn.myapplication.ServerModels.RasPiInversor;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class OtrosActivity extends AppCompatActivity {

    private Button btnSolar, btnRiego, btnPiscina; // A: Solar, B: Riego, C:
    Piscina, D: Otros
    private RasPiCtrlStat EstadoCtrl = new RasPiCtrlStat(false, false, false, false,
    false, false);
    private TextView TxtPresion, TxtDeposito, TxtNivel;
    final HTTPAsincrono IniciaHTTP = new HTTPAsincrono();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_otros);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        IniciaVariables();
        LeerEstado();
        SetTextEstado();
        IniciaHTTP.execute();

        btnSolar = findViewById(R.id.button_A_0);
        btnRiego = findViewById(R.id.button_B_0);
        btnPiscina = findViewById(R.id.button_C_0);
        btnSolar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CuentaAtras.Pulsaciones(true);
                Intent intent = new Intent(OtrosActivity.this, SolarActivity.class);
                startActivity(intent);
                overridePendingTransition(0,0);
                IniciaHTTP.cancel(true);
            }
        });
        btnRiego.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            CuentaAtras.Pulsaciones(true);
            Intent intent = new Intent(OtrosActivity.this, RiegoActivity.class);
            startActivity(intent);
            overridePendingTransition(0,0);
            IniciaHTTP.cancel(true);
        }
    });
    btnPiscina.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            CuentaAtras.Pulsaciones(true);
            Intent intent = new Intent(OtrosActivity.this, PiscinaActivity.class);
            startActivity(intent);
            overridePendingTransition(0,0);
            IniciaHTTP.cancel(true);
        }
    });

    findViewById(R.id.BtnAgua).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            CuadroDialogoCtrlAgua();
        }
    });
    findViewById(R.id.BtnAjustes).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            CuadroDialogoCFG();
        }
    });
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        // Esto es lo que hace mi botón al pulsar ir a atrás
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        return super.onKeyDown(keyCode, event);
    }
    return super.onKeyDown(keyCode, event);
}

private void CuadroDialogoCtrlAgua(){
    AlertDialog.Builder builder = new AlertDialog.Builder(OtrosActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_ctrl_agua, null);
    builder.setView(view);

    final AlertDialog dialog = builder.create();
    dialog.show();

    SetTextCtrlAgua(view);

    view.findViewById(R.id.BtnLlenar).setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        String Msg;
        if(EstadoCtrl.isLlenando()){
            //EstadoCtrl.setLlenando(false);
            //Enviar apagar llenado
            EscribeEstado("STL");
            Msg = "Parando llenado...";
        }
        else{

```



```

        //EstadoCtrl.setLlenando(true);
        //Enviar presion ON
        EscribeEstado("LLD");
        Msg = "Encendiendo llenado...";
    }
    Toast toast = Toast.makeText(getApplicationContext(),Msg,
Toast.LENGTH_SHORT);
    toast.show();
    SetTextEstado();
    dialog.dismiss();
    SetTextEstado();
}
});

view.findViewById(R.id.BtnPresion).setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        String Msg;
        if(EstadoCtrl.isPresion()){
            //EstadoCtrl.setPresion(false);
            //Enviar presion OFF
            EscribeEstado("PPR");
            Msg = "Apagando presión...";
        }
        else{
            //EstadoCtrl.setPresion(true);
            //Enviar presion ON
            EscribeEstado("ENP");
            Msg = "Encendiendo presión...";
        }
        Toast toast = Toast.makeText(getApplicationContext(),Msg,
Toast.LENGTH_SHORT);
        toast.show();
        SetTextEstado();
        dialog.dismiss();
        SetTextEstado();
    }
});

}

private void CuadroDialogoCFG(){
    AlertDialog.Builder builder = new AlertDialog.Builder(OtrosActivity.this,
R.style.DialogBasicCustomStyle);
    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.dialog_ajustes_auto, null);

    final ToggleButton LLAuto, PrAuto, Exced;
    LLAuto = view.findViewById(R.id.SWLLA);
    PrAuto = view.findViewById(R.id.SWPR);
    Exced = view.findViewById(R.id.SWEX);
    Exced.setEnabled(false);
    LLAuto.setChecked(EstadoCtrl.isLlenadoAuto());
    PrAuto.setChecked(EstadoCtrl.isPresionAuto());

    builder.setView(view).setPositiveButton("Enviar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Toast toast = Toast.makeText(getApplicationContext(),"Enviando
configuración...", Toast.LENGTH_SHORT);
            toast.show();
            if (LLAuto.isChecked()) EscribeEstado("CFL1");
            else EscribeEstado("CFLO");
            if (PrAuto.isChecked()) EscribeEstado("CFP1");
            else EscribeEstado("CFPO");
            //Código para enviar
        }
    }
}

```

```

    }).setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {

        }
    }).setCancelable(false);

    AlertDialog dialog = builder.create();
    dialog.show();
}

private void SetTextCtrlAgua(View view){
    String Text;
    if(EstadoCtrl.isPresion()) Text = "Apagar presión";
    else Text = "Encender presión";
    Button Presion = view.findViewById(R.id.BtnPresion);
    Presion.setText(Text);
    if(EstadoCtrl.isLlenando()) Text = "Parar llenado depósito";
    else Text = "Llenar depósito";
    Button Llenar = view.findViewById(R.id.BtnLlenar);
    Llenar.setText(Text);
}

private void IniciaVariables(){
    TxtPresion = findViewById(R.id.TextPresion);
    TxtDeposito = findViewById(R.id.TextStatusDeposito);
    TxtNivel = findViewById(R.id.TextNivel);
}

private void SetTextEstado(){
    String Text;
    //Nivel depósito
    if(EstadoCtrl.isLleno()){
        if(EstadoCtrl.isNoVacio()) Text = "- Nivel depósito: Lleno";
        else Text = "- Error en los sensores de nivel";
    }else{
        if (EstadoCtrl.isNoVacio()) Text = "- Nivel depósito: Medio";
        else Text = "- Nivel depósito: Vacío";
    }
    TxtNivel.setText(Text);
    //Estado depósito
    if(EstadoCtrl.isLlenando()) Text = "- Estado depósito: Llenando";
    else Text = "- Estado depósito: No se está llenando";
    TxtDeposito.setText(Text);

    //Estado presión
    if(EstadoCtrl.isPresion()) Text = "- Estado presión: Encendida";
    else Text = "- Estado presión: Apagada";
    TxtPresion.setText(Text);
}

private void LeerEstado(){ //Lee estado de la base de datos
    RasPiService service = RasPi.getRasPi().create(RasPiService.class);
    Call<RasPiCtrlStat> RasPiCall = service.viewStatus(RasPi.RasPiKey);
    RasPiCall.enqueue(new Callback<RasPiCtrlStat>() {
        @Override
        public void onResponse(Call<RasPiCtrlStat> call, Response<RasPiCtrlStat>
response) {
            EstadoCtrl = response.body();
            SetTextEstado();
        }

        @Override
        public void onFailure(Call<RasPiCtrlStat> call, Throwable t) {
            Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_SHORT).show();
        }
    });
}

private void EscribeEstado(String CMD){ //Escribe estado en la base de datos
    RasPiService service = RasPi.getRasPi().create(RasPiService.class);
    Call<Void> RasPiCall = service.newCtrlStatus(RasPi.RasPiKey, CMD);
}

```

```

RasPiCall.enqueue(new Callback<Void>() {
    @Override
    public void onResponse(Call<Void> call, Response<Void> response) {

    }

    @Override
    public void onFailure(Call<Void> call, Throwable t) {
        Toast.makeText(getApplicationContext(), "Error de conexión",
Toast.LENGTH_SHORT).show();
    }
});
}

private class HTTPAsincrono extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
        LeerEstado();
        //Toast.makeText(SolarActivity.this, "Actualizado",
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onCancelled(Void aVoid) {
        super.onCancelled(aVoid);
    }

    @Override
    protected void onCancelled() {
        super.onCancelled();
        //Toast.makeText(SolarActivity.this, "Cancelado",
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        int i = 0;
        while(!isCancelled() && i<600){
            publishProgress();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            i++;
        }
        return null;
    }
}
}

```

2.7.1.2.2. ChartModels

2.7.1.2.2.1. LineChartBaterly

```

package com.example.alexn.myapplication.ChartModels;
import android.content.Context;
import android.graphics.Color;
import android.graphics.Paint;

import com.example.alexn.myapplication.Ativities.MainActivity;
import com.example.alexn.myapplication.FnModels.ColorMPAChart;
import com.example.alexn.myapplication.FnModels.GridLines;
import com.example.alexn.myapplication.R;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.Legend;
import com.github.mikephil.charting.components.LegendEntry;
import com.github.mikephil.charting.components.LimitLine;
import com.github.mikephil.charting.components.XAxis;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.formatter.IndexAxisValueFormatter;
import com.github.mikephil.charting.utils.ColorTemplate;

import java.util.ArrayList;

public class LineChartBaterly {
    private LineChart linechart;
    private Context context;
    private ArrayList<Entry> Valores;
//-----
// Variables fijas
// Nombres de los colores principales, habrá que crearlos tras pasar el context -----
private int C_Linea, C_Relleno, C_Consumo, C_Produccion,
           C_EjeY, C_EjeX, C_Leyenda,
           C_GridLines, C_Limite; //Declaración de los colores en formato adecuado

// Variables importantes ya definidas -----
private static final int Alpha_Relleno = 180;
private static final float GrosorLinea = 1.5f,
                          SizeLetraY = 8f, SizeLetraX = 10f,
                          LimiteBateria = 65;
private LineDataSet set1 = CreateSet(ValoresSetLabels(), "");

//-----
// Métodos
public LineChartBaterly(){}

public LineChartBaterly(LineChart linechart, ArrayList<Entry> valores, Context
Context) { //, String[] valoresX
    this.linechart = linechart;
    Valores = valores;
    //ValoresX = valoresX;
    context = Context;
//-----
// COLORES DEL GRÁFICO
    C_Linea = new ColorMPAChart(R.color.ColorBateria, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Consumo = new ColorMPAChart(R.color.ColorConsumo, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Produccion = new ColorMPAChart(R.color.ColorProduccion, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Relleno = new ColorMPAChart(R.color.ColorBateria, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_EjeY = new ColorMPAChart(R.color.colorPrimaryDark, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_EjeX = new ColorMPAChart(R.color.colorPrimaryDark, ColorMPAChart.NOAlpha,
context).getColorMP();
}

```

```

        C_Leyenda = new ColorMPAChart(R.color.colorPrimaryDark, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_GridLines = new ColorMPAChart(R.color.colorPrimaryDark, 100,
context).getColorMP();
        C_Limite = new ColorMPAChart(R.color.colorLimite, 100, context).getColorMP();
    }

    public void SetData(){

        LineDataSet set2 = CreateSet(Valores, "Batería");
        LineData data = new LineData(set1, set2);
        linechart.setData(data);
        SetLineChartProperties();
        SetAxisYProperties();
        SetAxisXProperties(SetLabels());
        SetLegendProperties();

    }

    public ArrayList<Entry> NewArrayList(float[] Posiciones, float[] Valores){
        ArrayList<Entry> Lista = new ArrayList<>();
        for (int i = 0; i<Posiciones.length; i++){
            Lista.add(new Entry(Posiciones[i], Valores[i]));
        }
        return Lista;
    }

    public void SetData2(ArrayList<Entry> valores){
        LineDataSet set2 = CreateSet(valores, "");
        LineData data = new LineData(set1, set2);
        linechart.setData(data);
        SetLineChartProperties();
        SetAxisYProperties();
        SetAxisXProperties(SetLabels());
        SetLegendProperties();
    }

    public void NoData(){
        linechart.clear();
        linechart.setNoDataText("");
    }

    public void SetNewData(float[] Valores){}

    private LineDataSet CreateSet(ArrayList<Entry> yVals ,String Label){
        // create a dataset and give it a type -----
        ----- SET1: Batería. Para este tipo de gráfico sólo hay una entrada
        LineDataSet set;
        set= new LineDataSet(yVals, Label);
        set.setFillAlpha(Alpha_Relleno);
        set.setFillColor(C_Relleno);
        // set the line to be drawn like this "- - - - -" //
        set1.enableDashedLine(10f, 5f, 0f); // set1.enableDashedHighlightLine(10f, 5f, 0f);
        set.setColor(C_Linea);
        set.setCircleColor(C_Linea);
        set.setLineWidth(GrosorLinea);
        set.setDrawCircles(false); // set1.setCircleRadius(0f);
        set.setDrawCircleHole(false);
        set.setValueTextSize(0f);
        set.setDrawFilled(true);
        set.setMode(LineDataSet.Mode.LINEAR); //Splines cúbicos en lugar de rectas
        return set;
    }

    private void SetLineChartProperties(){
        linechart.setPinchZoom(false);
        linechart.setTouchEnabled(false);
        linechart.animateY(300);
        linechart.getDescription().setEnabled(false);
    }

    private void SetAxisYProperties(){
        //Eje Y Izquierda -----
        YAxis yLeft = linechart.getAxisLeft();
        yLeft.setAxisMaximum(108f);
        yLeft.setAxisMinimum(0f);
    }

```



```

        return yVals;
    }
    private ArrayList<String> SetLabels() {

        String[] Horas = new String[241];
        for (int i = 0; i < Horas.length; i++) {
            Horas[i] = ".";
        }
        for (int i = 0; i < Horas.length; i=i+10) {
            Horas[i] = ".";
        }
        for (int i = 0; i < Horas.length; i=i+40 ) {
            Horas[i]=Integer.toString(i/10)+"h";
        }
        ArrayList<String> xVals = new ArrayList<String>();
        for (int i = 0; i < Horas.length; i++) {
            xVals.add(Horas[i]);
        }
        return xVals;
    }

}

//
// Unused

public Context getContext() {
    return context;
}

public void setContext(Context context) {
    this.context = context;
}

public LineChartBaterly(LineChart linechart) {
    this.linechart = linechart;
}

public ArrayList<Entry> getValores() {
    return Valores;
}

public void setValores(ArrayList<Entry> valores) {
    Valores = valores;
}

public LineChart getLinechart() {
    return linechart;
}

public void setLinechart(LineChart linechart) {
    this.linechart = linechart;
}

}

```

2.7.1.2.2.2. LineChartPower

```

package com.example.alexn.myapplication.ChartModels;

import android.content.Context;

import com.example.alexn.myapplication.FnModels.ColorMPAChart;
import com.example.alexn.myapplication.FnModels.GridLines;
import com.example.alexn.myapplication.R;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.Legend;
import com.github.mikephil.charting.components.XAxis;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.formatter.IndexAxisValueFormatter;

import java.util.ArrayList;

import static
com.github.mikephil.charting.components.Legend.LegendDirection.RIGHT_TO_LEFT;

public class LineChartPower {
    private LineChart linechart;
    //private float[] ValoresY;
    //private String[] ValoresX;
    private Context context;
    private ArrayList<Entry> ValCons;
    private ArrayList<Entry> ValProd;

    //Nombres de los colores principales, habrá que crearlos tras pasar el context -----
    private int C_Linea, C_Linea_P, C_Linea_C, C_Bateria,
        C_Relleno, C_Relleno_P, C_Relleno_C,
        C_EjeY, C_EjeX,
        C Leyenda, C GridLines, C Limite
        ; //Declaración de los colores en formato adecuado

    //Variables importantes ya definidas -----
    private static final int Alpha_Relleno = 0, Alpha_Relleno_C = 20, Alpha_Relleno_P =
20;
    private static final float GrosorLinea = 2f;
    private static final float SizeLetraY = 8f;
    private static final float SizeLetraX = 12f;
    private static final float LimiteBateria = 65;
    private LineDataSet set1 = CreateSet(ValoresSetLabels(), "");

    //
    // Métodos
    public LineChartPower() {}

    public LineChartPower(LineChart linechart, ArrayList<Entry> valcons,
ArrayList<Entry> valprod, Context Context) { //, String[] valoresX
        this.linechart = linechart;
        ValCons = valcons;
        ValProd = valprod;
        context = Context;
    }

    ----- COLORES DEL GRÁFICO
    C_Linea = new ColorMPAChart(R.color.ColorProduccion, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Linea_P = new ColorMPAChart(R.color.ColorProduccion, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Linea_C = new ColorMPAChart(R.color.ColorConsumo, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Bateria = new ColorMPAChart(R.color.ColorBateria, ColorMPAChart.NOAlpha,
context).getColorMP();
    C_Relleno = new ColorMPAChart(R.color.ColorProduccion, ColorMPAChart.NOAlpha,
context).getColorMP();

```



```

        C_Relleno_P = new ColorMPAChart(R.color.ColorProduccion, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_Relleno_C = new ColorMPAChart(R.color.ColorConsumo, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_EjeY = new ColorMPAChart(R.color.colorWhite, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_EjeX = new ColorMPAChart(R.color.colorWhite, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_Leyenda = new ColorMPAChart(R.color.colorWhite, ColorMPAChart.NOAlpha,
context).getColorMP();
        C_GridLines = new ColorMPAChart(R.color.colorWhite, 200, context).getColorMP();
        C_Limite = new ColorMPAChart(R.color.colorLimite, 100, context).getColorMP();
    }

    public void SetData(){

        LineDataSet set2 = CreateSetCons(ValCons, "Consumo");
        LineDataSet set3 = CreateSetProd(ValProd, "Producción");
        LineData data = new LineData(set1, set2, set3);
        linechart.setData(data);
        SetLineChartProperties();
        SetAxisYProperties();
        SetAxisXProperties(SetLabels());
        SetLegendProperties();

    }

    public void NoData(){
        linechart.clear();
        linechart.setNoDataText("");
    }

    public void SetData2(ArrayList<Entry> consumo, ArrayList<Entry> produccion){
        LineDataSet set2 = CreateSetCons(consumo, "Consumo");
        LineDataSet set3 = CreateSetProd(produccion, "Producción");
        LineData data = new LineData(set1, set2, set3);
        linechart.setData(data);
        SetLineChartProperties();
        SetAxisYProperties();
        SetAxisXProperties(SetLabels());
        SetLegendProperties();
    }

    public ArrayList<Entry> NewArrayList(float[] Posiciones, float[] Valores){
        ArrayList<Entry> Lista = new ArrayList<>();
        for (int i = 0; i<Posiciones.length; i++){
            Lista.add(new Entry(Posiciones[i], Valores[i]));
        }
        return Lista;
    }

    private LineDataSet CreateSet(ArrayList<Entry> yVals ,String Label){
        // create a dataset and give it a type -----
        ----- SET1: Bateria. Para este tipo de gráfico sólo hay una entrada
        LineDataSet set;
        set= new LineDataSet(yVals, Label);
        set.setFillAlpha(Alpha_Relleno);
        set.setFillColor(C_Relleno);
        // set the line to be drawn like this "- - - - -" //
        set1.enableDashedLine(10f, 5f, 0f); // set1.enableDashedHighlightLine(10f, 5f, 0f);
        set.setColor(C_Linea);
        set.setCircleColor(C_Linea);
        set.setLineWidth(GrosorLinea);
        set.setDrawCircles(false); // set1.setCircleRadius(0f);
        set.setDrawCircleHole(false);
        set.setValueTextSize(0f);
        set.setDrawFilled(true);
        set.setMode(LineDataSet.Mode.LINEAR); //Splines cúbicos en lugar de rectas
        return set;
    }

    private LineDataSet CreateSetCons(ArrayList<Entry> yVals ,String Label){
        // create a dataset and give it a type -----
        ----- SET1: Bateria. Para este tipo de gráfico sólo hay una entrada
        LineDataSet set;

```

```

        set= new LineDataSet(yVals, Label);
        set.setFillAlpha(Alpha_Relleno_C);
        set.setFillColor(C_Relleno_C);
        // set the line to be drawn like this "- - - - -" //
set1.enableDashedLine(10f, 5f, 0f); // set1.enableDashedHighlightLine(10f, 5f, 0f);
        set.setColor(C_Linea_C);
        set.setCircleColor(C_Linea_C);
        set.setLineWidth(GrosorLinea);
        set.setDrawCircles(false); // set1.setCircleRadius(0f);
        set.setDrawCircleHole(false);
        set.setValueTextSize(0f);
        set.setDrawFilled(true);
        set.setMode(LineDataSet.Mode.LINEAR); //Splines cúbicos en lugar de rectas
        return set;
    }
    private LineDataSet CreateSetProd(ArrayList<Entry> yVals ,String Label){
        // create a dataset and give it a type -----
        ----- SET1: Batería. Para este tipo de gráfico sólo hay una entrada
        LineDataSet set;
        set= new LineDataSet(yVals, Label);
        set.setFillAlpha(Alpha_Relleno_P);
        set.setFillColor(C_Relleno);
        // set the line to be drawn like this "- - - - -" //
set1.enableDashedLine(10f, 5f, 0f); // set1.enableDashedHighlightLine(10f, 5f, 0f);
        set.setColor(C_Linea);
        set.setCircleColor(C_Linea);
        set.setLineWidth(GrosorLinea);
        set.setDrawCircles(false); // set1.setCircleRadius(0f);
        set.setDrawCircleHole(false);
        set.setValueTextSize(0f);
        set.setDrawFilled(true);
        set.setMode(LineDataSet.Mode.LINEAR); //Splines cúbicos en lugar de rectas
        return set;
    }
    private void SetLineChartProperties(){
        linechart.setPinchZoom(false);
        linechart.setTouchEnabled(false);
        linechart.animateY(300);
        linechart.getDescription().setEnabled(false);
    }
    private void SetAxisYProperties(){
        //Eje Y Izquierda -----
        float A = linechart.getData().getDataSetByIndex(1).getYMax();
        float Max = linechart.getData().getDataSetByIndex(2).getYMax();
        if (A>Max){ Max = A; }
        YAxis yLeft = linechart.getAxisLeft();
        yLeft.setAxisMaximum(Max*1.1f);
        yLeft.setAxisMinimum(-Max*0.02f);
        yLeft.setDrawAxisLine(false);
        yLeft.setLabelCount(2);
        yLeft.setTextColor(C_EjeY);
        yLeft.setTextSize(SizeLetraY);
        yLeft.setDrawGridLines(true);
        yLeft.setGridColor(C_GridLines);
        yLeft.setDrawLabels(false);
        yLeft.setPosition(YAxis.YAxisLabelPosition.OUTSIDE_CHART);
        yLeft.setCenterAxisLabels(true);
        yLeft.setDrawGridLines(false);

        //Lineas de referencias importantes en el gráfico -----
        //yLeft.addLimitLine(new GridLines(LimiteBateria, C_GridLines, C_Limite, false,
true, true).getLimite());
        yLeft.removeAllLimitLines();
        yLeft.addLimitLine(new GridLines(Max, C_GridLines, C_GridLines, true, false,
true, false).getLimite());
        yLeft.addLimitLine(new GridLines(0, C_GridLines, C_GridLines, true, false, true,
false).getLimite());
        yLeft.addLimitLine(new GridLines(Max*0.25f, C_GridLines, C_GridLines, true,
false, true, false).getLimite());
        yLeft.addLimitLine(new GridLines(Max*0.5f, C_GridLines, C_GridLines, true,
false, true, false).getLimite());
    }

```



```
// Unused

public Context getContext() {
    return context;
}

public void setContext(Context context) {
    this.context = context;
}

public LineChartPower(LineChart linechart) {
    this.linechart = linechart;
}

public ArrayList<Entry> getValCons() {
    return ValCons;
}

public void setValCons(ArrayList<Entry> valores) {
    ValCons = valores;
}

public LineChart getLinechart() {
    return linechart;
}

public void setLinechart(LineChart linechart) {
    this.linechart = linechart;
}
}
```

2.7.1.2.3. FnModels

2.7.1.2.3.1. FormatoFecha

```

package com.example.alexnm.myapplication.FnModels;

import android.media.TimedText;
import android.text.format.Time;
import android.text.style.TtsSpan;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Locale;

public class FormatoFecha {
    String Fecha;

    public static String getCurrentDate() {
        String Fecha;
        Date date = new Date();
        Fecha = DateFormat.getDateInstance(DateFormat.FULL).format(date);
        Fecha = Fecha.substring(0,1).toUpperCase() + Fecha.substring(1);
        return Fecha;
    }

    public static String AnyDaySolar(int year, int month, int day) {
        String Fecha;
        Calendar cal = NewCalendar(year, month, day);
        Date date = cal.getTime();
        Fecha = DateFormat.getDateInstance(DateFormat.FULL).format(date);
        Fecha = Fecha.substring(0,1).toUpperCase() + Fecha.substring(1);
        return Fecha;
    }

    public static String getHour() {
        Date date = new Date(); // given date
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        calendar.setTime(date); // assigns calendar to given date
        String Time;
        int Hora = calendar.get(Calendar.HOUR_OF_DAY); // gets hour in 24h format
        if (Hora < 10) Time = "0"+Hora;
        else Time = ""+Hora;
        int Min = calendar.get(Calendar.MINUTE); // gets month number, NOTE this
is zero based!
        if (Min < 10) Time = Time + ":0" + Min;
        else Time = Time + ":" + Min;
        return Time;
    }

    public static int GetH(){
        Date date = new Date(); // given date
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        calendar.setTime(date); // assigns calendar to given date
        String Time;
        return calendar.get(Calendar.HOUR_OF_DAY); // gets hour in 24h format
    }

    public static int GetMin(){
        Date date = new Date(); // given date
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        calendar.setTime(date); // assigns calendar to given date
        String Time;
        return calendar.get(Calendar.MINUTE); // gets min
    }
}

```

```

    public static int GetYear(){
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        return calendar.get(Calendar.YEAR);
    }

    public static int GetMonth(){
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        return (calendar.get(Calendar.MONTH)+1);
    }

    public static int GetMonthDay(){
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        return calendar.get(Calendar.DAY_OF_MONTH);
    }

    public static String getDate(){
        //Date date = new Date(); // given date
        Calendar calendar = GregorianCalendar.getInstance(); // creates a new calendar
instance
        //calendar.setTime(date); // assigns calendar to given date
        String Day = calendar.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.getDefault());
        Day = Day.substring(0,1).toUpperCase() + Day.substring(1);
        int Dia = calendar.get(Calendar.DAY_OF_MONTH);
        int Mes = calendar.get(Calendar.MONTH) + 1;
        int Ano = calendar.get(Calendar.YEAR);
        return Day + ", " + Dia + "/" + Mes + "/" + Ano;
    }

    public static String AnyDayFecha(int year, int month, int day){
        String Fecha;
        Calendar cal = NewCalendar(year, month, day);
        String Day = cal.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.getDefault());
        Day = Day.substring(0,1).toUpperCase() + Day.substring(1);
        return Day + ", " + day + "/" + month + "/" + year;
    }

    public static Calendar NewCalendar(int year, int month, int day){
        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.YEAR, year);
        cal.set(Calendar.MONTH, month -1);
        cal.set(Calendar.DAY_OF_MONTH, day);
        cal.set(Calendar.HOUR_OF_DAY, 0);
        cal.set(Calendar.MINUTE, 0);
        cal.set(Calendar.SECOND, 0);
        cal.set(Calendar.MILLISECOND, 0);
        return cal;
    }

    public static String AnyDayFechaHist(int year, int month, int day){
        String Fecha;
        Calendar cal = NewCalendar(year, month, day);
        String Day = cal.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.getDefault());
        Day = Day.substring(0,1).toUpperCase() + Day.substring(1);
        return Day + "\n" + day + "/" + month + "/" + year;
    }
}
}

```

2.7.1.2.3.2. GridLines

```

package com.example.alexnm.myapplication.FnModels;

import com.github.mikephil.charting.components.LimitLine;

public class GridLines {
    private LimitLine Limite;
    private static final float TextSize = 10;

    public GridLines(float posicion, int ColorT, int ColorL, boolean TOP,
boolean LIMIT, boolean Text, boolean Bateria) {
        String Position = "";

        if (Text){
            if (Bateria){
                Position = Integer.toString((int)posicion) + "%";
            }
            else{
                if ((posicion - (int)posicion) == 0){
                    Position = Integer.toString((int)posicion) + " W";
                }else{
                    Position = Float.toString(posicion) + " W";
                }
            }
        }
        LimitLine Lim = new LimitLine(posicion, Position);
        if (LIMIT) {
            //Limit line máximo
            Lim.setLineWidth(1f);
            Lim.enableDashedLine(10f, 10f, 0f);
            if (TOP) {
                Lim.setLabelPosition(LimitLine.LimitLabelPosition.LEFT_TOP);
            }else{
                Lim.setLabelPosition(LimitLine.LimitLabelPosition.LEFT_BOTTOM);
            }
            Lim.setTextSize(TextSize);
            Lim.setLineColor(ColorL);
            Lim.setTextColor(ColorT);
        } else {
            //Grid line
            Lim.setLineWidth(0.3f);
            Lim.enableDashedLine(10f, 0f, 0f);
            if (TOP) {
                Lim.setLabelPosition(LimitLine.LimitLabelPosition.LEFT_TOP);
            } else {
                Lim.setLabelPosition(LimitLine.LimitLabelPosition.LEFT_BOTTOM);
            }
            Lim.setTextSize(TextSize);
            Lim.setLineColor(ColorL);
            Lim.setTextColor(ColorT);
        }
        Limite = Lim;
    }

    public LimitLine getLimite() {
        return Limite;
    }

    public void setLimite(LimitLine limite) {
        Limite = limite;
    }
}

```

2.7.1.2.4. RasPiServer

2.7.1.2.4.1. RasPiService

```

package com.example.alexn.myapplication.RasPiServer.RasPiServices;

import com.example.alexn.myapplication.ServerModels.RasPiCtrlStat;
import com.example.alexn.myapplication.ServerModels.RasPiData;
import com.example.alexn.myapplication.ServerModels.RasPiInversor;
import com.example.alexn.myapplication.ServerModels.RasPiTask;
import com.example.alexn.myapplication.ServerModels.RasPiValores;

import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Query;

public interface RaspPiService {

    @GET("pruebaserver.php")
    Call<RasPiData> UpdateValuesPrueba(@Query("pwd") String Password,
    @Query("action") String Accion, @Query("table") String Tabla, @Query("Valor1")
    String Valor1, @Query("Valor2") String Valor2);

    @GET("pruebaserver.php")
    Call<RasPiValores> LeeTablaFull(@Query("pwd") String Password,
    @Query("action") String Accion, @Query("table") String Tabla);

    @GET("pruebaserver.php")
    Call<RasPiValores> LeeTablaFecha(@Query("pwd") String Password,
    @Query("action") String Accion, @Query("table") String Tabla, @Query("fecha")
    String Fecha);

    @GET("DataInversor.php")
    Call<RasPiInversor> LeeInversorFecha(@Query("pwd") String Password,
    @Query("Day") String Day, @Query("Month") String Month, @Query("Year") String
    Year, @Query("Last") String Last, @Query("LastID") String LastID);

    @GET("ClearTask.php")
    Call<Void> ClearTaskt(@Query("pwd") String Password, @Query("TASK") String
    Task, @Query("DELETE") String Delete);

    @GET("NewTask.php")
    Call<Void> NewTaskt(@Query("pwd") String Password, @Query("Tipo") String
    Tipo, @Query("Comando") String Comando, @Query("Day") String Day,
    @Query("Month") String Month, @Query("Year") String Year, @Query("H") String
    H, @Query("Min") String Min);

    @GET("ViewTask.php")
    Call<RasPiTask> ViewTask(@Query("pwd") String Password, @Query("TASK")
    String Task);

    @GET("ViewStatus.php")
    Call<RasPiCtrlStat> ViewStatus(@Query("pwd") String Password);

    @GET("NewCtrlStatus.php")
    Call<Void> NewCtrlStatus(@Query("pwd") String Password, @Query("CMD")
    String CMD);
}

```


2.7.1.2.4.2. RasPi

```

package com.example.alexn.myapplication.RasPiServer;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RasPi {

    public static final String BASE_URL =
"http://myhuertoapp.duckdns.org/";
    public static final String RasPiKey = "*****";
    //Contraseña páginas php
    public static final String RasPiSave = "SAVE";
    public static final String RasPiRead = "READ";

    private static Retrofit retrofit = null;

    public static Retrofit getRasPi(){
        if (retrofit == null){
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)

.addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }

    //Método para formatear la fecha
    public String GetFecha(String Day, String Month, String Year){
        String Fecha = Year + "-" + Month + "-" + Day + "%";
        return Fecha;
    }

    //Método para formatear la fecha con horas
    public String GetHour(String Day, String Month, String Year,
String Hour){
        String Fecha = Year + "-" + Month + "-" + Day + " "+ Hour +
":%";
        return Fecha;
    }

    //Método para formatear la fecha con minutos
    public String GetMinutes(String Day, String Month, String Year,
String Hour, String Min){
        String Fecha = Year + "-" + Month + "-" + Day + " "+ Hour +
":" + Min + "%";
        return Fecha;
    }
}

```

2.7.1.2.5. ServerModels

2.7.1.2.5.1. RasPiCtrlStat

```

package com.example.alexn.myapplication.ServerModels;

public class RasPiCtrlStat {
    boolean Lleno, NoVacio, Llenando, Presion, LlenadoAuto, PresionAuto;

    public RasPiCtrlStat() {
    }

    public RasPiCtrlStat(boolean lleno, boolean noVacio, boolean llenando, boolean
presion, boolean llenadoAuto, boolean presionAuto) {
        Lleno = lleno;
        NoVacio = noVacio;
        Llenando = llenando;
        Presion = presion;
        LlenadoAuto = llenadoAuto;
        PresionAuto = presionAuto;
    }

    public boolean isLleno() {
        return Lleno;
    }

    public void setLleno(boolean lleno) {
        Lleno = lleno;
    }

    public boolean isNoVacio() {
        return NoVacio;
    }

    public void setNoVacio(boolean noVacio) {
        NoVacio = noVacio;
    }

    public boolean isLlenando() {
        return Llenando;
    }

    public void setLlenando(boolean llenando) {
        Llenando = llenando;
    }

    public boolean isPresion() {
        return Presion;
    }

    public void setPresion(boolean presion) {
        Presion = presion;
    }

    public boolean isLlenadoAuto() {
        return LlenadoAuto;
    }

    public void setLlenadoAuto(boolean llenadoAuto) {
        LlenadoAuto = llenadoAuto;
    }

    public boolean isPresionAuto() {
        return PresionAuto;
    }

    public void setPresionAuto(boolean presionAuto) {
        PresionAuto = presionAuto;
    }
}

```

2.7.1.2.5.2. RasPiData

```
package com.example.alexn.myapplication.ServerModels;

import com.google.gson.annotations.Expose;

public class RasPiData {
    @Expose
    private int Id;
    private float Valor1, Valor2;
    private String Hora;

    public RasPiData() {}

    public RasPiData(int id, float valor1, float valor2, String hora)
    {
        Id = id;
        Valor1 = valor1;
        Valor2 = valor2;
        Hora = hora;
    }

    public int getId() {
        return Id;
    }

    public float getValor1() {
        return Valor1;
    }

    public float getValor2() {
        return Valor2;
    }

    public String getHora() {
        return Hora;
    }

    public void setId(int id) {
        Id = id;
    }

    public void setValor1(float valor1) {
        Valor1 = valor1;
    }

    public void setValor2(float valor2) {
        Valor2 = valor2;
    }

    public void setHora(String hora) {
        Hora = hora;
    }
}
```

2.7.1.2.5.3. RasPiInversor

```

package com.example.alexn.myapplication.ServerModels;

public class RasPiInversor {

    private int[] Id;
    private String[] Hora;
    private float[] Pos, VA, WC, Bat, Temp, WP;
    private int LastID;
    private float LastBat, VAh, WCh, WPh;

    public RasPiInversor(){}

    public RasPiInversor(int[] id, String[] hora, float[] Pos, float[] VA, float[] WC,
float[] bat, float[] temp, float[] WP, int lastID, float lastBat, float VAh, float WCh,
float WPh) {
        Id = id;
        Hora = hora;
        this.VA = VA;
        this.WC = WC;
        Bat = bat;
        Temp = temp;
        this.WP = WP;
        LastID = lastID;
        LastBat = lastBat;
        this.VAh = VAh;
        this.WCh = WCh;
        this.WPh = WPh;
    }

    public int[] getId() {
        return Id;
    }

    public void setId(int[] id) {
        Id = id;
    }

    public String[] getHora() {
        return Hora;
    }

    public void setHora(String[] hora) {
        Hora = hora;
    }

    public float[] getPos() {
        return Pos;
    }

    public void setPos(float[] pos) {
        Pos = pos;
    }

    public float[] getVA() {
        return VA;
    }

    public void setVA(float[] VA) {
        this.VA = VA;
    }

    public float[] getWC() {
        return WC;
    }

    public void setWC(float[] WC) {
        this.WC = WC;
    }

    public float[] getBat() {
        return Bat;
    }
}

```

```

    }

    public void setBat(float[] bat) {
        Bat = bat;
    }

    public float[] getTemp() {
        return Temp;
    }

    public void setTemp(float[] temp) {
        Temp = temp;
    }

    public float[] getWP() {
        return WP;
    }

    public void setWP(float[] WP) {
        this.WP = WP;
    }

    public int getLastID() {
        return LastID;
    }

    public void setLastID(int lastID) {
        LastID = lastID;
    }

    public float getLastBat() {
        return LastBat;
    }

    public void setLastBat(float lastBat) {
        LastBat = lastBat;
    }

    public float getVAh() {
        return VAh;
    }

    public void setVAh(float VAh) {
        this.VAh = VAh;
    }

    public float getWCh() {
        return WCh;
    }

    public void setWCh(float WCh) {
        this.WCh = WCh;
    }

    public float getWPh() {
        return WPh;
    }

    public void setWPh(float WPh) {
        this.WPh = WPh;
    }
}

```

2.7.1.2.5.4. RasPiMonitorInv

```

package com.example.alexn.myapplication.ServerModels;

import com.github.mikephil.charting.data.Entry;

import java.util.ArrayList;
import java.util.Collection;

public class RasPiMonitorInv {
    private String HoraInicial;
    private float ValorInicial;
    private String HoraFinal;
    private float ValorFinal;
    private float[] Posicion, ValBat, ValCons, ValProd;
    private float BatAVG, ProdAVG, ConsAVG;

    public RasPiMonitorInv() {}

    public ArrayList<Entry> GetBateria() {
        ArrayList<Entry> Bateria = new ArrayList<>();
        for (int i = 0; i < Posicion.length; i++) {
            Bateria.add(new Entry(Posicion[i], ValBat[i]));
        }
        return Bateria;
    }

    public ArrayList<Entry> GetConsumo() {
        ArrayList<Entry> Consumo = new ArrayList<>();
        for (int i = 0; i < Posicion.length; i++) {
            Consumo.add(new Entry(Posicion[i], ValCons[i]));
        }
        return Consumo;
    }

    public ArrayList<Entry> GetProduccion() {
        ArrayList<Entry> Produccion = new ArrayList<>();
        for (int i = 0; i < Posicion.length; i++) {
            Produccion.add(new Entry(Posicion[i], ValProd[i]));
        }
        return Produccion;
    }
}

```

2.7.1.2.5.5. RasPiTask

```
package com.example.alexnm.myapplication.ServerModels;

public class RasPiTask {
    private String Tareas, Cabecera;

    public RasPiTask() {
    }

    public RasPiTask(String tareas, String cabecera) {
        Tareas = tareas;
        Cabecera = cabecera;
    }

    public String getTareas() {
        return Tareas;
    }

    public void setTareas(String tareas) {
        Tareas = tareas;
    }

    public String getCabecera() {
        return Cabecera;
    }

    public void setCabecera(String cabecera) {
        Cabecera = cabecera;
    }
}
```

2.7.1.2.5.6. RasPiValores

```
package com.example.alexn.myapplication.ServerModels;

public class RasPiValores {
    private int[] ID;
    private float[] Valor1, Valor2;
    private String[] Hora;

    public RasPiValores () {}

    public RasPiValores(int[] id, float[] valor1, float[] valor2,
String[] hora) {
        this.ID = id;
        Valor1 = valor1;
        Valor2 = valor2;
        Hora = hora;
    }

    public int[] getId() {
        return ID;
    }

    public float[] getValor1() {
        return Valor1;
    }

    public float[] getValor2() {
        return Valor2;
    }

    public String[] getHora() {
        return Hora;
    }

    public void setId(int[] id) {
        this.ID = id;
    }

    public void setValor1(float[] valor1) {
        Valor1 = valor1;
    }

    public void setValor2(float[] valor2) {
        Valor2 = valor2;
    }

    public void setHora(String[] hora) {
        Hora = hora;
    }
}
```


2.7.2. ESP32

2.7.2.1. Zona finca

Se compone de 11 archivos de código:

- Archivo principal: ZonaFinca.ino
 - Variables y definiciones
 - Setup
 - Main

- 00_Setup.ino
 - Definiciones
 - Funciones de Setup generales

- 01_Fn_Principal.ino
 - Funciones para el programa principal

- 02_Fn_Comprueba_Máquinas.ino
 - Subfunciones de la rutina ciclo máquina

- 03_Maquinas.ino
 - Funciones máquina de estados

- 04_Fn_RiegoDeposito.ino
 - Funciones de cada estado de la máquina de estados y de setup de la misma

- 05_Fn_Depuradora.ino
 - Setup depuradora

- 06_Fn_Presion.ino
 - Setup presión

- 07_Interrupciones.ino
 - Timer
 - LoRa
 - Función de envío LoRa

- 08_Fn_Datos_INT.ino
 - Funciones de conversión de valores

- 09_Fn_Orden_INT.ino
 - Funciones de actuación de órdenes.

2.7.2.1.1. Archivo principal

```

#include <LoRa.h>
#include <SPI.h>
#include <Wire.h>
#include "SSD1306.h"
#define RST_LED 16 // GPIO16 -- OLED reset pin
#define SDA 4 // GPIO4 -- SX127x's SDA
#define SCL 15 // GPIO15 -- SX127X's SCL
SSD1306 display(0x3c, SDA, SCL, RST_LED);

//Pines para entradas/salidas
int DO_BP = 22; int DO_BD = 21; int DO_BPr = 13; int DO_EVA = 25; int
DO_EVB = 33; int DO_EVD = 32; int DI_NL = 36; int DI_NV = 37;
// ESTADO SENSORES/ACTUADORES
//Estado sensores
bool T_Pozo, T_Depuradora, T_Presion, N_Vacio, N_Lleno;
//Estado actuadores
bool B_Pozo, B_Depuradora, B_Presion, EV_A, EV_B, EV_D;

bool ErrorInv;
int Contador, Sum_CVA, Sum_CW, Sum_PBat, Sum_TInv, Sum_PW, Consumo;
bool R_A, R_B, RegandoA, RegandoB, L_D, LlenandoD, Llenado_Auto;
int EstadoRiegoDep;
int TimEspera;
bool Depurar, Depurando;
bool Presion_Auto, Presion;
bool RST_System;
String Tarea;

#define unseg 80 //Valor de tiempo correspondiente a 1 segundo
hw_timer_t * timer1 = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

bool T1S, SENDLORA;
int T_LoRa, T_A, T_B, T_D, T_ON_BP, T_ON_BD;

void IRAM_ATTR onTimer1() {
    TimersAction();
}

void setup() {
    display.init();
    display.flipScreenVertically();
    display.setFont(ArialMT_Plain_10);
    display.clear();
    SerialSetup();
    LoRaSetup();
    DigitalIOSetup();
    InitVariables();
    Timer1Setup(1);
}

```

```
void loop() {
  T1S = LOW;
  int Timeout = 0;
  int TimeErrorInv = millis();
  while(Serial.available() < 100) {
    ActualizaOLED();
    if(Timeout < millis()) Serial.print("QPIGS\xb7\xa9\x0d"); //Timeout
    cumplido
    if(Timeout < millis() || (millis()+1000) < Timeout) Timeout = millis()
    + 1000; //Timeout cumplido o desbordamiento de la variables que almacena
    los milisegundos
    if ((millis() - TimeErrorInv) > 3000) ErrorInv = HIGH; //3 segundos
    sin respuesta
    Maquina();
    if (SENDLORA) {
      SendLoRa();
      InitInversor();
      T_LoRa = 0;
      SENDLORA = LOW;
    }
  }
  LeeInversor();
  ErrorInv = LOW;
  while(!T1S) { ActualizaOLED(); }
}
```

2.7.2.1.2. 00_Setup

```

#define SCK      5    // GPIO5  -- SX127x's SCK
//#define SDA    4    // GPIO4  -- SX127x's SDA
//#define SCL    15   // GPIO15 -- SX127X's SCL
#define MISO     19   // GPIO19 -- SX127x's MISO
#define MOSI     27   // GPIO27 -- SX127x's MOSI
#define SS       18   // GPIO18 -- SX127x's CS
#define RST      14   // GPIO14 -- SX127x's RESET
#define DI00     26   // GPIO26 -- SX127x's IRQ(Interrupt Request)
//#define RST_LED 16   // GPIO16 -- OLED reset pin
#define BAND     868E6 //433E6 //you can set band here directly,e.g.
868E6,915E6
#define PABOOST true

void SerialSetup(){
    Serial.begin(2400);
}

void LoRaSetup(){
    SPI.begin(SCK,MISO,MOSI,SS);
    LoRa.setPins(SS,RST,DI00);
    if (!LoRa.begin(BAND,PABOOST)) { Serial.println("Error LoRa"); while
(1);}
    LoRa.onReceive(onReceive);
    LoRa.receive();
}

void DigitalIOSetup(){
    //Pines para controlar entradas y salidas
    pinMode(DO_BP, OUTPUT);
    digitalWrite(DO_BP, LOW);
    pinMode(DO_BD, OUTPUT);
    digitalWrite(DO_BD, LOW);
    pinMode(DO_BPr, OUTPUT);
    digitalWrite(DO_BPr, LOW);
    pinMode(DO_EVA, OUTPUT);
    digitalWrite(DO_EVA, LOW);
    pinMode(DO_EVB, OUTPUT);
    digitalWrite(DO_EVB, LOW);
    pinMode(DO_EVD, OUTPUT);
    digitalWrite(DO_EVD, LOW);
    pinMode(DI_NL, INPUT);
    N_Lleno = digitalRead(DI_NL);
    pinMode(DI_NV, INPUT);
    N_Vacio = digitalRead(DI_NV);
}

```

```

void InitVariables(){
    B_Pozo = LOW; B_Depuradora = LOW; B_Presion = LOW; EV_A = LOW; EV_B =
LOW; EV_D = LOW;
    ErrorInv = LOW;
    SENDLORA = LOW;
    T1S = LOW;
    T_A = 0; T_B = 0; T_D = 0; T_ON_BP = 0; T_ON_BD = 0; T_LoRa = 0;
    WriteOutputs();
    ReadInputs();
    SetupRiegoDeposito();
    SetupPresion();
    SetupDepuradora();
    InitInversor();
}

void InitInversor(){
    Contador = 0;
    Sum_CVA = 0;
    Sum_CW = 0;
    Sum_PBat = 0;
    Sum_TInv = 0;
    Sum_PW = 0;
}

void Timer1Setup(int TimSeg){
    timer1 = timerBegin(1, unseg*TimSeg, true); //80 en el segundo campo
es 1 segundo periódicamente
    timerAttachInterrupt(timer1, &onTimer1, true);
    timerAlarmWrite(timer1, 1000000, true);
    timerAlarmEnable(timer1);
}

```

2.7.2.1.3. 01_FnPrincipal

```

void Maquina() {
    ReadInputs();
    Estados_Presion();
    Estados_Depuradora();
    Estados_Riego_Deposito();
    if(RST_System) ReseteaSistema();
    WriteOutputs();
}

void LeeInversor() {
    int Aparente, Bateria, Temperatura, Produccion;
    Contador++; //Debe inicializarse a 0 cada vez que se envíe un dato
    String ReadSerial = Serial.readString(); //Utilizar Serial2 en el
inversor de verdad
    Aparente = ReadSerial.substring(24,28).toInt();
    Sum_CVA = Sum_CVA + Aparente;
    Consumo = ReadSerial.substring(29,33).toInt();
    Sum_CW = Sum_CW + Consumo;
    Bateria = ReadSerial.substring(52,55).toInt();
    Sum_PBat = Sum_PBat + Bateria;
    Temperatura = ReadSerial.substring(56,60).toInt();
    Sum_TInv = Sum_TInv + Temperatura;
    Produccion = ReadSerial.substring(99,104).toInt();
    Sum_PW = Sum_PW + Produccion;
    //No es la forma más precisa de calcular el valor medio cada minuto,
pero es la más sencilla y el error va a ser pequeño debido a que son
muestras regulares
}

void ActualizaOLED() {
    display.clear();
    float pctj = T_LoRa*100/60;
    int P = (int)pctj;
    display.drawString(0, 0, "Esperando envio: " + String(P,DEC) + "%");
    display.drawString(0, 10, "Ultima orden: " + Tarea);
    if(ErrorInv) display.drawString(0, 20, "Error Inversor!");
    display.display();
}

```

2.7.2.1.4. 02_Fn_Comprueba_Maquinas

```

void ReadInputs() {
    N_Lleno = digitalRead(DI_NL);
    N_Vacio = digitalRead(DI_NV);
    T_Pozo = !ErrorInv && (T_ON_BP > 10) && (Consumo < 1000);
    T_Depuradora = !ErrorInv && (T_ON_BD > 10) && (Consumo < 1000);
    T_Presion = LOW; //Debido al funcionamiento de esta bomba no es
práctico estimar de esta forma un fallo por relé de sobrecarga, además no
es una bomba crítica para el funcionamiento del sistema
    if (!ErrorInv && (T_ON_BP > 10) && (T_ON_BD > 10) && (Consumo < 2000)){
        T_Pozo = HIGH;
        T_Depuradora = HIGH;
    }
    if(N_Lleno) L_D = LOW;
    if(T_A == 0) R_A = LOW;
    if(T_B == 0) R_B = LOW;
    if(T_D == 0) Depurar = LOW;
}

void WriteOutputs() {
    //B_Pozo = LOW; B_Depuradora = LOW; B_Presion = LOW; EV_A = LOW; EV_B =
LOW; EV_D = LOW;
    digitalWrite(DO_BP, B_Pozo);
    digitalWrite(DO_BD, B_Depuradora);
    digitalWrite(DO_BPr, B_Presion);
    digitalWrite(DO_EVA, EV_A);
    digitalWrite(DO_EVB, EV_B);
    digitalWrite(DO_EVD, EV_D);
}

void ReseteaSistema() {
    InitVariables();
    RST_System = LOW;
}

```

2.7.2.1.5. 03_Maquinas

```
void Estados_Riego_Deposito(){
  if(Llenado_Auto && !N_Lleno) L_D = HIGH;
  switch(EstadoRiegoDep){
    //Estado inicial
    case 0: Pozo_Est_0();
    break;
    case 1: Pozo_Est_1();
    break;
    case 2: Pozo_Est_2();
    break;
    case 3: Pozo_Est_3();
    break;
    case 4: Pozo_Est_4();
    break;
    case 5: Pozo_Est_5();
    break;
    case 100: Pozo_Est_100();
    break;
    default: EstadoRiegoDep = 0;
    break;
  }
}

void Estados_Depuradora(){
  if(T_Depuradora){
    Depurar = LOW;
    T_D = 0;
    T_ON_BD = 0;
    Depurando = LOW;
  }else{
    B_Depuradora = Depurar;
    Depurando = Depurar;
  }
}

void Estados_Presion(){
  if(Presion_Auto) B_Presion = N_Vacio;
  else B_Presion = Presion;
}
```


2.7.2.1.6. 04_Fn_RiegoDeposito

```

void SetupRiegoDeposito(){
  R_A= LOW; R_B = LOW; RegandoA = LOW; RegandoB = LOW; L_D = LOW;
  LlenandoD = LOW; Llenado_Auto = LOW;
  EstadoRiegoDep = 0;
}

void Pozo_Est_0(){
  B_Pozo = LOW;
  EV_A = LOW;
  EV_B = LOW;
  EV_D = LOW;
  RegandoA = LOW;
  RegandoB = LOW;
  LlenandoD = LOW;
  T_ON_BP = 0;
  if(R_A||R_B||L_D) EstadoRiegoDep = 1;
}

void Pozo_Est_1(){
  //Estado timeout espera 1
  TimEspera = millis() + 1000;
  EstadoRiegoDep = 2;
}

void Pozo_Est_2(){
  //Estado abre primera válvula
  B_Pozo = LOW;
  if (R_A) EV_A = HIGH;
  if (R_B) EV_B = HIGH;
  if (L_D) EV_D = HIGH;
  RegandoA = LOW;
  RegandoB = LOW;
  LlenandoD = LOW;
  if (millis() < (TimEspera-1000)) TimEspera = millis() + 1000;
  if (millis() > TimEspera) EstadoRiegoDep = 3;
}

void Pozo_Est_3(){
  //Estado ON
  B_Pozo = HIGH;
  if (R_A && !EV_A) EV_A = HIGH;
  if (R_B && !EV_B) EV_B = HIGH;
  if (L_D && !EV_D) EV_D = HIGH;
  if (!R_A && (R_B || L_D)) EV_A = LOW;
  if (!R_B && (R_A || L_D)) EV_B = LOW;
  if (!L_D && (R_A || R_B)) EV_D = LOW;
  RegandoA = EV_A;
  RegandoB = EV_B;
  LlenandoD = EV_D;
  if(T_Pozo) EstadoRiegoDep = 100;
  else if(!R_A && !R_B && !L_D) EstadoRiegoDep = 4;
}

```

```
void Pozo_Est_4() {
    TimEspera = millis() + 1000;
    EstadoRiegoDep = 5;
}

void Pozo_Est_5() {
    B_Pozo = LOW;
    RegandoA = LOW;
    RegandoB = LOW;
    LlenandoD = LOW;
    T_ON_BP = 0;
    //Estado para hacer SET del timer
    if (millis() < (TimEspera-1000)) TimEspera = millis() + 1000;
    if (millis() > TimEspera) EstadoRiegoDep = 0;
}

void Pozo_Est_100() {
    R_A = LOW;
    R_B = LOW;
    L_D = LOW;
    T_A = 0;
    T_B = 0;
    T_D = 0;
    EstadoRiegoDep = 4;
}
```

2.7.2.1.7. 05_Fn_Depuradora

```
void SetupDepuradora() {  
    Depurar = LOW; Depurando = LOW;  
}
```

2.7.2.1.8. 06_Fn_Presion

```
void SetupPresion() {  
    Presion_Auto = LOW;  
}
```

2.7.2.1.9. 07_Interrupciones

```

void TimersAction(){
  T1S = HIGH;
  if (Depurando){
    T_ON_BD++;
    if(T_D > 0) T_D --; }
  if(RegandoA || RegandoB ||LlenandoD) T_ON_BP++;
  if(RegandoA & T_A > 0) T_A--;
  if(RegandoB & T_B > 0) T_B--;
  T_LoRa++;
  if(T_LoRa > 58) SENDLORA = HIGH;
}

void SendLoRa () {
  display.clear();
  display.drawString(0, 0, "Enviando LoRa...");
  display.display();
  if(ErrorInv) {Sum_CVA = 0; Sum_CW = 0; Sum_PBat = 0; Sum_TInv = 0;
Sum_PW=0; Contador = 1;}
  LoRa.beginPacket();
  LoRa.write(HighByte(ShortLoRa(Sum_CVA, Contador)));
  LoRa.write(LowByte(ShortLoRa(Sum_CVA, Contador)));
  LoRa.write(HighByte(ShortLoRa(Sum_CW, Contador)));
  LoRa.write(LowByte(ShortLoRa(Sum_CW, Contador)));
  LoRa.write(ByteLoRa(Sum_PBat, Contador));
  LoRa.write(ByteLoRa(Sum_TInv, Contador));
  LoRa.write(HighByte(ShortLoRa(Sum_PW, Contador)));
  LoRa.write(LowByte(ShortLoRa(Sum_PW, Contador)));
  //Escribir bytes de estado
  LoRa.write(HighByte8_9());
  LoRa.write(LowByte8_9());
  LoRa.write(Byte10());
  LoRa.write(Byte11());
  LoRa.endPacket();
  ResetValuesInv();
  LoRa.receive();
}

void onReceive(int packetSize){
  if (packetSize <0 || packetSize > 4) return;
  String Orden;
  byte bX;
  for (int i = 0; i<3; i++){
    Orden += (char)LoRa.read();
  }
  if (packetSize == 4){
    bX = LoRa.read();
    int X = bX;
    Orden4(Orden, X);
    Tarea = Orden + String(X,DEC);
    //Serial.println(Orden);
    return;
  }
  Orden3(Orden);
  Tarea = Orden;
  //Serial.println(Orden);
}

```

2.7.2.1.10. 08_Fn_Datos_INT

```

void ResetValuesInv() {
    Contador = 0;
    Sum_CVA = 0;
    Sum_CW = 0;
    Sum_PBat = 0;
    Sum_TInv = 0;
    Sum_PW = 0;
}

short ShortLoRa(int Suma, int contador){
    short RetShort = 0;
    float Media = (float)Suma / (float)contador;
    float DecimalMedia = Media - (int)Media;
    RetShort = Media;
    RetShort = RetShort << 2; //Los dos bits bajos quedarán a 0, son los
    decimales = Num,0
    if (DecimalMedia >= 0.125 && DecimalMedia<0.375) RetShort |= 0x0001;
    else if (DecimalMedia >= 0.375 && DecimalMedia<0.625) RetShort |=
    0x0002;
    else if (DecimalMedia >= 0.625 && DecimalMedia<0.875) RetShort |=
    0x0003;
    else if (DecimalMedia >= 0.875){
        RetShort = Media + 1;
        RetShort = RetShort << 2;
    }
    //Serial.println(RetShort);
    return RetShort;
}

byte ByteLoRa(int Suma, int contador){
    byte RetByte = 0;
    float Media = (float)Suma / (float)contador;
    float DecimalMedia = Media - (int)Media;
    RetByte = Media;
    RetByte = RetByte << 1; //Los dos bits bajos quedarán a 0, son los
    decimales = Num,0
    if (DecimalMedia >= 0.25 && DecimalMedia<0.75) RetByte |= 0x0001;
    else if (DecimalMedia >= 0.75){
        RetByte = Media + 1;
        RetByte = RetByte << 1;
    }
    return RetByte;
}

byte LowByte(short Value){
    byte B = Value;
    return B;
}

byte HighByte(short Value){
    Value = Value >> 8;
    byte B = Value;
    return B;
}

```

```
byte LowByte8_9(){
    byte B = 0;
    if (ErrorInv) B |= 0x08;
    if (T_Pozo) B |= 0x40;
    if (T_Depuradora) B |= 0x80;
    return B;
}

byte HighByte8_9(){
    byte B = 0;
    if (T_Presion) B |= 0x01;
    if (N_Lleno) B |= 0x02;
    if (N_Vacio) B |= 0x04;
    return B;
}

byte Byte10(){
    //bool B_Pozo, B_Depuradora, B_Presion, EV_A, EV_B, EV_D;
    byte B = 0;
    if (B_Pozo) B |= 0x01;
    if (B_Depuradora) B |= 0x02;
    if (B_Presion) B |= 0x04;
    if (EV_A) B |= 0x08;
    if (EV_B) B |= 0x10;
    if (EV_D) B |= 0x20;
    return B;
}

byte Byte11(){
    byte B = 0;
    if (Llenado_Auto) B |= 0x01;
    if (Presion_Auto) B |= 0x02;
    if (RegandoA) B |= 0x04;
    if (RegandoB) B |= 0x08;
    if (LlenandoD) B |= 0x10;
    if (Depurando) B |= 0x20;
    return B;
}
```

2.7.2.1.11. 09_Fn_Orden:INT

```

void Orden4(String Orden, int X){
  if (Orden == "RRA"){ //Regar lado A: X min
    R_A = HIGH;
    T_A = X*60; return;
  }
  if (Orden == "RRB"){ //Regar lado B: X min
    R_B = HIGH;
    T_B = X*60; return;
  }
  if (Orden == "RAB"){ //Regar lado A y B: X min
    R_A = HIGH;
    R_B = HIGH;
    T_A = X*60;
    T_B = X*60; return;
  }
  if (Orden == "DEP"){ //Depurar: X min
    Depurar = HIGH;
    T_D = X*60; return;
  }
  if (Orden == "CFL"){
    if (X == 1) Llenado_Auto = HIGH;
    else Llenado_Auto = LOW; return;
  }
  if (Orden == "CFP"){
    if (X == 1) Presion_Auto = HIGH;
    else Presion_Auto = LOW; return;
  }
}

void Orden3(String Orden){
  if (Orden == "LLD" && !N_Lleno){ //llenarDeposito
    L_D = HIGH; return;
  }
  if (Orden == "STR"){ //Stop riego
    R_A = LOW;
    R_B = LOW;
    T_A = 0;
    T_B = 0; return;
  }
  if (Orden == "STD"){ //Stop depuradora
    Depurar = LOW;
    T_D = 0; return;
  }
  if (Orden == "STL"){ //Stop llenado
    L_D = LOW; return;
  }
  if (Orden == "ENP"){ //Encender presión
    Presion = HIGH; return;
  }
  if (Orden == "PPR"){ //Parar presión
    Presion = LOW; return;
  }
  if (Orden == "RST"){ //llenarDeposito
    RST_System = HIGH; return;
  }
}

```

2.7.2.2. Zona vivienda

Se compone de 6 archivos de código:

- Principal
 - Setup
 - Main
- Funciones
 - Funciones generales
- Funciones_WiFi
 - Funciones HTTP para enviar y recibir datos del servidor
- Interrupción_Timer
 - Funciones de interrupción timer
- LoRa
 - Funciones para enviar y recibir LoRa
- Setup
 - Funciones para setup

2.7.2.2.1. Principal

```

#include <WiFiClient.h>
#include <HTTPClient.h>
#include <WiFiAP.h>
#include <Wire.h>
#include "SSD1306.h"
#define RST_LED 16 // GPIO16 -- OLED reset pin
#define SDA 4 // GPIO4 -- SX127x's SDA
#define SCL 15 // GPIO15 -- SX127X's SCL
SSD1306 display(0x3c, SDA, SCL, RST_LED);
#define unseg 80 //Valor de tiempo correspondiente a 1
segundo
hw_timer_t * timer1 = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
bool INTIME, MSGLoRa, FirstMSG, ErrorCOMLoRa, NewTask, Orden4;
short TIME;
String DatosLoRa, PaqueteLoRa, Tarea;
byte X;
byte B[12];
int TIEMPO;
void IRAM_ATTR onTimer1() {
    Timer1s();
}
void setup() {
    Serial.begin(9600);
    display.init();
    display.setFont(ArialMT_Plain_10);
    LoRaSetup();
    for (int i = 1; !WiFiSetup(i); i++) if (i>4) i = 1;
    VariableSetup();
    Timer1Setup(1);
}
void loop() {
    if(TIEMPO != TIME) Actualiza_OLED();
    if(MSGLoRa){
        LoRaDATA();
        display.clear();
        display.drawString(0, 0, "Recibido!");
        display.drawString(0, 10, "GUARDANDO DATOS...");
        display.display();
        NewData(); INTIME = HIGH; MSGLoRa = LOW;
    }
    if (INTIME){
        PendingTask();
        if(NewTask) SendLoRa();
    }else if(ErrorCOMLoRa){
        DatosLoRa = "&ErrCOMLoRa=1";
        NewData();
        display.clear();
        display.drawString(0, 0, "ERROR DE");
        display.drawString(0, 10, "COMUNICACIÓN!");
        display.drawString(0, 20, "Esperando LoRa...");
        display.display();
        while(ErrorCOMLoRa){delay(1000);}
    }
}

```

2.7.2.2.2. Funciones

```
String LeeWord(byte H, byte L){
    short WL = 0;
    short WH = 0;
    WL = L;
    WH = H << 8;
    short W = WL | WH;
    float Dato = (float)W/4;
    return String(Dato, 2);
}

String LeeByte(byte B){
    float Dato = (float)B/2;
    return String(Dato, 1);
}

String wordToString(byte H, byte L){
    short WL = 0;
    short WH = 0;
    WL = L;
    WH = H << 8;
    short W = WL | WH;
    return String((int)W);
}

String byteToString(byte B){
    return String((int)B);
}

void Actualiza_OLED(){
    display.clear();
    display.drawString(0, 0, "WiFi Conectado!");
    TIEMPO = TIME;
    display.drawString(0, 10, "Tiempo: " + String(TIEMPO,DEC) + "s ");
    if(INTIME) display.drawString(0, 20, "In Time!");
    else display.drawString(0, 20, "Esperando LoRa...");
    display.display();
}
```

2.7.2.2.3. Funciones_WiFi

```

void PendingTask() {
    if (WiFi.status() != WL_CONNECTED) for (int i = 1; !WiFiSetup(i); i++)
if (i>4) i = 1;
    HTTPClient http;
    http.begin("http://myhuertoapp.duckdns.org/PendingTask.php?pwd=*****
*****");
    int httpCode = http.GET();
    if (httpCode <= 0) {
        NewTask = LOW;
        return;
    }
    String payload = http.getString();
    http.end();
    Tarea = (payload.substring(0,3));
    if (payload.length()>3) {
        X = payload.substring(3).toInt();
        Orden4 = HIGH;
        Serial.print(Tarea);
        Serial.println(X);
    }else{
        X=0;
        Orden4 = LOW;
    }
    if(Tarea.length()>=3) NewTask = HIGH;
    else NewTask = LOW;
}

void NewData() {
    if (WiFi.status() != WL_CONNECTED) for (int i = 1; !WiFiSetup(i); i++)
if (i>4) i = 1;
    HTTPClient http;
    String urlFija =
"http://myhuertoapp.duckdns.org/NewData.php?pwd=*****";
    String url = urlFija + DatosLoRa;
    http.begin(url);
    int httpCode = http.GET();
    if (httpCode <= 0) return;
    String payload = http.getString();
    http.end();
}

```

2.7.2.2.4. Interrupción_Timer

```

void Timer1s(){
    TIME++;
    if(TIME > 57) INTIME = LOW;
    if(TIME > 65) ErrorCOMLoRa = HIGH;
}

```

2.7.2.2.5. LoRa

```

void onReceive(int packetSize){
    if(packetSize != 12) return;
    FirstMSG = HIGH;
    for (int i = 0; i < packetSize; i++) {
        B[i] = LoRa.read();
    }
    MSGLoRa = HIGH;
    TIME = 0;
    ErrorCOMLoRa = LOW;
}

void LoRaDATA(){
    DatosLoRa = "&VA=" + LeeWord(B[0], B[1]) +
                "&WC=" + LeeWord(B[2], B[3]) +
                "&Bat=" + LeeByte(B[4]) +
                "&Temp=" + LeeByte(B[5]) +
                "&WP=" + LeeWord(B[6], B[7]) +
                "&WordSens=" + wordToString(B[8], B[9]) +
                "&ByteAct=" + byteToString(B[10]) +
                "&Estado=" + byteToString(B[11])+
                "&ErrCOMLoRa=0";
}

void SendLoRa () {
    LoRa.beginPacket ();
    LoRa.print(Tarea);
    if (Orden4) LoRa.write(X);
    LoRa.endPacket ();
    LoRa.receive ();
}

```

2.7.2.2.6. Setup

```

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include "SSD1306.h"
#define SCK      5      // GPIO5  -- SX127x's SCK
#define SDA      4      // GPIO4  -- SX127x's SDA
#define SCL     15     // GPIO15 -- SX127X's SCL
#define MISO     19     // GPIO19 -- SX127x's MISO
#define MOSI     27     // GPIO27 -- SX127x's MOSI
#define SS       18     // GPIO18 -- SX127x's CS
#define RST      14     // GPIO14 -- SX127x's RESET
#define DI00     26     // GPIO26 -- SX127x's IRQ(Interrupt Request)
#define RST_LED  16     // GPIO16 -- OLED reset pin
#define BAND     868E6 //433E6 //you can set band here directly,e.g.
868E6,915E6
#define PABOOST  true
#define LED      25

const char* ssid1 = "MOVISTAR_9B61"; const char* password1 =
"*****"; //WiFi Casa 1
const char* ssid2 = "ONOC34"; const char* password2 =
"*****"; //WiFi Casa 2
const char* ssid3 = "Honor10"; const char* password3 =
"*****"; //WiFi Movil

String SSIDs[] = {ssid1, ssid2, ssid3};
String Pass[] = {password1, password2, password3};

void LoRaSetup() {
  display.clear();
  display.drawString(0, 0, "Iniciando LoRa...");
  display.display();
  SPI.begin(SCK,MISO,MOSI,SS);
  LoRa.setPins(SS,RST,DI00);
  if (!LoRa.begin(BAND,PABOOST)) {
    display.clear();
    display.drawString(0, 0, "ERROR INICIO LORA!");
    display.display();
    while (1);
  }
  //LoRa.setSpreadingFactor(10);
  //LoRa.setSignalBandwidth(125E3);
  LoRa.onReceive(onReceive);
  LoRa.receive();
}

void Timer1Setup(int TimSeg){
  timer1 = timerBegin(1, unseg*TimSeg, true); //80 en el segundo campo
es 1 segundo periódicamente
  timerAttachInterrupt(timer1, &onTimer1, true);
  timerAlarmWrite(timer1, 1000000, true);
  timerAlarmEnable(timer1);
}

```

```

bool WiFiSetup(int i){
  pinMode(LED, OUTPUT);
  bool STATE = LOW;
  int Timeout = millis() + 5000;
  switch(i){
    case 1: WiFi.begin(ssid1, password1);
            display.clear();
            display.drawString(0, 0, "Conectando WiFi...");
            display.drawString(0, 10, "MOVISTAR_9B61");
            display.display();
            break;
    case 2: WiFi.begin(ssid2, password2);
            display.clear();
            display.drawString(0, 0, "Conectando WiFi...");
            display.drawString(0, 10, "ONOC34");
            display.display();
            break;
    case 3: WiFi.begin(ssid3, password3);
            display.clear();
            display.drawString(0, 0, "Conectando WiFi...");
            display.drawString(0, 10, "Honor10");
            display.display();
            break;
    default: WiFi.begin(ssid2, password2);
             display.clear();
             display.drawString(0, 0, "Conectando WiFi...");
             display.drawString(0, 10, " ONOC34");
             display.display();
             break;
  }
  //WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    digitalWrite(LED, STATE);
    STATE = !STATE;
    if (millis()+5000 < Timeout) Timeout = millis() + 5000;
    if (millis() > Timeout) return false;
  }
  digitalWrite(LED, HIGH);
  display.clear();
  display.drawString(0, 0, "WiFi Conectado!");
  display.display();
  return true;
}

void VariableSetup(){
  FirstMSG = LOW;
  INTIME = LOW;
  TIME = 0;
  NewTask = LOW;
}

```

2.7.3. PHP

La API programada en php consta de los siguientes archivos:

- HAppConfig.php
 - Contiene la información de conexión con la base de datos
- ClearTask.php
 - Borra tareas pendientes y programadas
- DataInversor.php
 - Devuelve en JSON los datos del inversor
- NewCtrlStatus.php
 - Actualiza variables de la tabla de estados
- NewData.php
 - Actualiza datos estado e inversor
- NewTask.php
 - Crea tarea
- PendingTask.php
 - Devuelve próxima orden a enviar
- ViewStatus.php
 - Devuelve variables de estado en JSON para mostrar en la app
- ViewTask.php
 - Devuelve la tarea formateada en JSON para la aplicación Android

2.7.3.1. *HAppConfig*

```
<?php
// config.php
// Credenciales
$dbhost = "localhost";
$dbuser = "huertoapp";
$dbpass = "*****";
$dbname = "HuertoApp";

// Conexión con la base de datos
$con = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
?>
```


2.7.3.2. ClearTask

```

<?php
require("HAppConfig.php");
$pwd = "*****";
// Contraseña
if ($_GET['pwd'] != $pwd){
    echo '<p>Acceso denegado</p>';
    die(); }
if ($_GET['TASK'] == "Riego")$Condition = "(Comando LIKE 'RRA%' OR
Comando LIKE 'RRB%' OR Comando LIKE 'RAB%')";
else if ($_GET['TASK'] == "Depuradora")$Condition = "Comando LIKE
'DEP%';
else $Condition = "1";
if ($_GET['DELETE'] == "Today"){
    $fecha = getdate();
    $yyyy = $fecha["year"];
    $mm = $fecha["mon"];
    if ($mm<10) $mm = "0".$mm;
    $dd = $fecha["mday"];
    if ($dd<10) $dd = "0".$dd;
    $hoy = $fecha["wday"];
    //Añade las tareas cíclicas a la tabla de enviadas, para que
no las envíe hoy
    $Q1 = "SELECT * FROM TareasCiclicas WHERE Activo = 1 AND Dia =
$hoy AND $Condition";
    $Ciclicas = mysqli_query($con, $Q1);
    while($NuevaTC = mysqli_fetch_array($Ciclicas)){
        $HoraTC = $NuevaTC["Hora"];
        $FechaTC = $yyyy."-".$mm."-".$dd;
        $Q1 = "INSERT INTO TCEnviada (Fecha, HoraTC)
VALUES ('$FechaTC', '$HoraTC')";
        mysqli_query($con, $Q1);
    }
    //Elimina las tareas pendientes de hoy
    $Q1 = "DELETE FROM TareasCola WHERE PendienteEnvio = 1 AND
FechaHora LIKE '$FechaTC%' AND $Condition";
    mysqli_query($con, $Q1);
    echo $Q1;
}
if ($_GET['DELETE'] == "All"){
    $Q1 = "DELETE FROM TareasCiclicas WHERE $Condition";
    mysqli_query($con, $Q1); //Elimina las tareas cíclicas
    $Q1 = "DELETE FROM TareasCola WHERE PendienteEnvio = 1 AND
$Condition";
    mysqli_query($con, $Q1); //Elimina las tareas pendientes de
hoy
}
$Q1 = "SELECT * FROM TareasCiclicas";
$Ciclicas = mysqli_query($con, $Q1);
if(mysqli_fetch_array($Ciclicas) == null){
    $Q1 = "TRUNCATE TABLE TareasCiclicas";
    mysqli_query($con, $Q1);
    echo $Q1; }
?>

```

2.7.3.3. *DataInversor*

```

<?php
require("HAppConfig.php");
$pwd = "*****";
// Contraseña
if ($_GET['pwd'] != $pwd){
    echo '<p>Acceso denegado</p>';
    die();
}
//Lee datos inversor de un día y los deja en un array
$Day = $_GET['Day'];
if ($Day < 10) $Day = "0".$Day;
$Month = $_GET['Month'];
if ($Month < 10) $Month = "0".$Month;
$Fecha = $_GET['Year']."-".$Month."-".$Day."%";
$query = "SELECT * FROM Inversor WHERE FechaHora LIKE '$Fecha'";
$DatosInv = mysqli_query($con, $Query);
$i = 0;
$VAh = 0;
$WCh = 0;
$WPh = 0;
while($Fila = mysqli_fetch_array($DatosInv)){
    $ArID[$i] = $Fila["Id"];
    $ArHora[$i] = substr($Fila["FechaHora"], -8);
    $ArPos[$i] = (substr($ArHora[$i], 0,2)*10 +
substr($ArHora[$i], 3,2)/6);
    $ArVA[$i] = $Fila["P_Aparente"];
    $ArWC[$i] = $Fila["P_Activa"];
    $ArBat[$i] = $Fila["P_Bateria"];
    $ArTe[$i] = $Fila["T_Inversor"];
    $ArWP[$i] = $Fila["P_Producida"];

    if ($i > 0){
        $DiffHora = //Calcula la diferencia en horas de la
primera a la última muestra
        (substr($ArHora[$i], 0,2) + substr($ArHora[$i], 3,2)/60 +
substr($ArHora[$i], 6,2)/3600) -
        (substr($ArHora[$i-1], 0,2) + substr($ArHora[$i-1],
3,2)/60 + substr($ArHora[$i-1], 6,2)/3600);
        $VAh = $VAh + (($ArVA[$i]+$ArVA[$i-1])/2)*($DiffHora);
        $WCh = $WCh + (($ArWC[$i]+$ArWC[$i-1])/2)*($DiffHora);
        $WPh = $WPh + (($ArWP[$i]+$ArWP[$i-1])/2)*($DiffHora);
    }
    $i++;
}
$i--;

```

```

        //Último Id para saber desde donde consultar parcialmente de ser
necesario
        $LastID = $ArID[$i];
        //Último valor del porcentaje de la batería
        $LastBat = $ArBat[$i];

        if($_GET['Last'] == "True"){
            if ($_GET['LastID'] == $LastID){
                $ArID = array(0);
                $ArHora = array("0");
                $ArPos = array(0);
                $ArVA = array(0);
                $ArWC = array(0);
                $ArBat = array(0);
                $ArTe = array(0);
                $ArWP = array(0);
                $LastID = 0;
            }
        }
        //Genera el JSON
        $datajson = json_encode(['Id'=> $ArID, 'Hora' => $ArHora,
'Pos'=>$ArPos, 'VA' => $ArVA, 'WC' => $ArWC, 'Bat' => $ArBat, 'Temp' =>
$ArTe, 'WP' => $ArWP, 'LastID' => $LastID,
'LastBat'=>$LastBat, 'VAh'=>$VAh,
'WCh'=>$WCh, 'WPh'=>$WPh]);
        echo $datajson;
    ?>

```

2.7.3.4. *NewCtrlStatus*

```
<?php
require ("HAppConfig.php");
$pwd = "*****";

// Contraseña -----
-----

if ($_GET['pwd'] != $pwd) {
    echo '<p>Acceso denegado</p>';
    die();
}

$Comando = $_GET['CMD'];
echo $Comando;

$Q1 = "INSERT INTO TareasCola (Comando, PendienteEnvio) VALUES
('$Comando', '1')";
mysqli_query($con, $Q1);

?>
```

2.7.3.5. *NewData*

```

<?php
    require("HAppConfig.php");
    $pwd = "*****";

    // Contraseña
    if ($_GET['pwd'] != $pwd){
        echo '<p>Acceso denegado</p>';
        die();
    }

    $VA = $_GET['VA'];
    $WC = $_GET['WC'];
    $BA = $_GET['Bat'];
    $TE = $_GET['Temp'];
    $WP = $_GET['WP'];
    $WSens = $_GET['WordSens'];
    $BAct = $_GET['ByteAct'];
    $ES = $_GET['Estado'];
    $ESCfg = $ES & 3;
    $RegA = ($ES&4) > 0;
    $RegB = ($ES&8) > 0;
    $Dep = ($ES&32) > 0;
    $LD = ($ES&16) > 0;
    $ErrorCOMLoRa = $_GET['ErrCOMLoRa'];

    if((($WSens&8) == 0) && $ErrorCOMLoRa == 0){
        $Q1 = "INSERT INTO Inversor (P_Aparente, P_Activa, P_Bateria,
T_inversor, P_Producida) VALUES('$VA', '$WC', '$BA', '$TE', '$WP')";
        echo $Q1;
        echo '<p></p>';
        mysqli_query($con, $Q1);
    }
    $Q1 = "INSERT INTO Estado (WordSensores, ByteActuadores, EstadoCfg,
RegandoA, RegandoB, Depurando, LlenandoD, ErrorCOMLoRa) VALUES('$WSens',
'$BAct', '$ESCfg', '$RegA', '$RegB', '$Dep', '$LD', '$ErrorCOMLoRa')";
    echo $Q1;
    mysqli_query($con, $Q1);
?>

```

2.7.3.6. NewTask

```

<?php
require("HAppConfig.php");
$pwd = "*****";
// Contraseña
if ($_GET['pwd'] != $pwd){
    echo '<p>Acceso denegado</p>';
    die();
}
$Comando = $_GET['Comando'];

if ($_GET['Tipo'] == "Unico"){
    $Fecha = $_GET['Year']."-".$_GET['Month']."-".$_GET['Day']."
".$_GET['H'].":".$_GET['Min'].":00";
    $Q1 = "INSERT INTO TareasCola (FechaHora, Comando, PendienteEnvio) VALUES
('$Fecha', '$Comando', '1')";
    mysqli_query($con, $Q1);
}else{
    $Hora = $_GET['H'].":".$_GET['Min'];
    $Days = decbin($_GET['Tipo']);
    $Days = "00000000".$Days;
    echo $Days;
    if (substr($Days, -1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '1', '$Comando', '1')"; //Lunes
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -2,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '2', '$Comando', '1')"; //Martes
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -3,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '3', '$Comando', '1')"; //Miércoles
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -4,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '4', '$Comando', '1')"; //Jueves
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -5,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '5', '$Comando', '1')"; //Viernes
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -6,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '6', '$Comando', '1')"; //Sábado
        mysqli_query($con, $Q1);
    }
    if (substr($Days, -7,1) == 1){
        $Q1 = "INSERT INTO TareasCiclicas (Hora, Dia, Comando, Activo)
VALUES ('$Hora', '0', '$Comando', '1')"; //Domingo
        mysqli_query($con, $Q1);
    }
}
?>

```

2.7.3.7. PendingTask

```

<?php
require("HAppConfig.php");
$pwd = "*****";
// Contraseña
if ($_GET['pwd'] != $pwd){
    echo '<p>Acceso denegado</p>';
    die();
}
//Obtiene datos de fecha para las consultas
$fecha = getdate();
$yyyy = $fecha["year"];
$mm = $fecha["mon"];
if ($mm<10) $mm = "0".$mm;
$dd = $fecha["mday"];
if ($dd<10) $dd = "0".$dd;
$hoy = $fecha["wday"];
$HoraActual = $fecha["hours"].":".$fecha["minutes"].":".$fecha["seconds"];

//Consulta TCEnviada para saber que tareas ciclicas se han enviado hoy
$Q1 = "SELECT * FROM TCEnviada WHERE Fecha = '$yyyy-$mm-$dd' ORDER BY HoraTC
DESC";
$query = mysqli_query($con, $Q1);
$DatosTC = mysqli_fetch_array($query);
$HMax = $DatosTC["HoraTC"];
//Consulta las tareas ciclicas pendientes de hoy con el dato de hora
obtenido de la anterior y cuya hora sea menor a la actual
$Q1 = "SELECT * FROM TareasCiclicas WHERE Activo = 1 AND Dia = $hoy AND Hora
> '$HMax' AND Hora <= '$HoraActual' ORDER BY Hora ASC";
$query = mysqli_query($con, $Q1);
//Añade todas las que cumplan la condición a tareas pendientes
while($NuevaTC = mysqli_fetch_array($query)){
    $Comando = $NuevaTC["Comando"];
    $HoraTC = $NuevaTC["Hora"];
    $FechaTC = $yyyy."-".$mm."-".$dd;
    $FechaHora = $FechaTC." ".$HoraTC;
    //Añade tarea a TareasCola
    $Q1 = "INSERT INTO TareasCola (FechaHora, Comando, PendienteEnvio)
VALUES('$FechaHora', '$Comando', '1')";
    mysqli_query($con, $Q1);
    //Añade tarea a TCEnviada
    $Q1 = "INSERT INTO TCEnviada (Fecha, HoraTC) VALUES('$FechaTC',
'$HoraTC')";
    mysqli_query($con, $Q1);
}
$FechaHora = $yyyy."-".$mm."-".$dd." ".$HoraActual;
//Consulta cola de tareas con la condicion de que no esten pendientes de
envio y la hora sea menor que la actual
$Q1 = "SELECT * FROM TareasCola WHERE FechaHora LIKE '$yyyy-$mm-$dd%' AND
FechaHora <= '$FechaHora' AND PendienteEnvio = '1' ORDER BY FechaHora ASC";
$query = mysqli_query($con, $Q1);
$NuevaT = mysqli_fetch_array($query);
echo $NuevaT["Comando"];
$IdSend = $NuevaT["Id"];
$Q1 = "UPDATE TareasCola SET PendienteEnvio = '0' WHERE Id =
'$IdSend'";
mysqli_query($con, $Q1);

```

```

$Q1 = "SELECT * FROM Estado ORDER BY Id DESC";
$Estados = mysqli_query($con, $Q1);
$EstadoActual = mysqli_fetch_array($Estados);

//Actualiza estado

$BAct = $EstadoActual["ByteActuadores"]; $WSens =
$EstadoActual["WordSensores"]; $EstCfg = $EstadoActual["EstadoCfg"];
$RegA = $EstadoActual["RegandoA"]; $RegB = $EstadoActual["RegandoB"];
$Dep = $EstadoActual["Depurando"]; $LD = $EstadoActual["LlenandoD"];
$errorCOMLoRa = $EstadoActual["ErrorCOMLoRa"];

if ($NuevaT["Comando"] != null) {
    switch (substr($NuevaT["Comando"], 0, 3)) {
        case "RRA": $RegA = 1; //Regar A
        break;
        case "RRB": $RegB = 1; //Regar B
        break;
        case "RAB": $RegA = 1; $RegB = 1; //Regar todo
        break;
        case "DEP": $Dep = 1; //Depurar
        break;
        case "CFL": $LA = substr($NuevaT["Comando"], 3, 1);
        //Configurar llenado auto
        if($LA == "1") $EstCfg = $EstCfg | 1;
        else $EstCfg = $EstCfg & 2;
        break;
        case "CFP": $PA = substr($NuevaT["Comando"], 3, 1);
        //Configurar presión auto
        if($PA == "1") $EstCfg = $EstCfg | 2;
        else $EstCfg = $EstCfg & 1;
        break;
        case "LLD": $LD = 1; //Llenar depósito
        break;
        case "STR": $RegA = 0; $RegB = 0; //Parar riego
        break;
        case "STL": $LD = 0; //Parar llenado deposito
        break;
        case "ENP": $BAct = $BAct | 4; //Encender presión
        break;
        case "PPR": $BAct = $BAct & 251; //Parar presión
        break;
        default:
    }
    $Q1 = "INSERT INTO Estado (ByteActuadores, WordSensores, EstadoCfg,
    RegandoA, RegandoB, Depurando, LlenandoD, ErrorCOMLoRa) VALUES('$BAct', '$WSens',
    '$EstCfg', '$RegA', '$RegB', '$Dep', '$LD', '$ErrorCOMLoRa')";
    mysqli_query($con, $Q1);
}
?>

```


2.7.3.8. *ViewStatus*

```

<?php
require("HAppConfig.php");
$pwd = "*****";

// Contraseña -----
-----

if ($_GET['pwd'] != $pwd) {
    echo '<p>Acceso denegado</p>';
    die();
}

$Q1 = "SELECT * FROM Estado ORDER BY Id DESC";
$Estados = mysqli_query($con, $Q1);
$EstadoActual = mysqli_fetch_array($Estados);
$WordSensores =
"0000000000000000".decbin($EstadoActual["WordSensores"]);
$Lleno = substr($WordSensores, -10, 1) == "1";
$NoVacio = substr($WordSensores, -11, 1) == "1";
$Llenando = $EstadoActual["LlenandoD"] == "1";
$Presion =
substr("00000000".decbin($EstadoActual["ByteActuadores"]), -3, 1) == "1";
$EstadoCfg = "00".decbin($EstadoActual["EstadoCfg"]);
$LlenadoAuto = substr($EstadoCfg, -1, 1) == "1";
$PresionAuto = substr($EstadoCfg, -2, 1) == "1";

if ($EstadoActual["WordSensores"] & 64) $Llenando = false;

//echo substr($WordSensores, -13, 1);

$datajson = json_encode(['Lleno'=>$Lleno, 'NoVacio'=>$NoVacio,
'Llenando'=>$Llenando, 'Presion'=>$Presion,
'LlenadoAuto'=>$LlenadoAuto, 'PresionAuto'=>$PresionAuto]);

echo $datajson;

?>

```

2.7.3.9. ViewTask

```

<?php
require("HAppConfig.php");
$pwd = "*****";

function LeeComandoDiaHora($comando, $dia, $hora){
    $cmd = substr($comando, 0,3);
    $duracion = substr($comando, 3);
    $horas = floor($duracion/60);
    $minutos = $duracion - $horas*60;
    $duracion = $horas."h y ".$minutos."min";
    $Partel = "";
    if($cmd == "RRA") $Partel = " - .numtodia($dia)." a las
".substr($hora, 0, 5)." \n          ·Regar lado A\n          ·Duración: ".$duracion."\n";
    if($cmd == "RRB") $Partel = " - .numtodia($dia)." a las
".substr($hora, 0, 5)." \n          ·Regar lado B\n          ·Duración: ".$duracion."\n";
    if($cmd == "RAB") $Partel = " - .numtodia($dia)." a las
".substr($hora, 0, 5)." \n          ·Regar todo\n          ·Duración: ".$duracion."\n";
    if($cmd == "DEP") $Partel = " - .numtodia($dia)." a las
".substr($hora, 0, 5)." \n          ·Depurar\n          ·Duración: ".$duracion."\n";
    return $Partel;
}

function LeeComandoDiaHora2($comando, $dia, $hora, $fecha){
    $cmd = substr($comando, 0,3);
    $duracion = substr($comando, 3);
    $horas = floor($duracion/60);
    $minutos = $duracion - $horas*60;
    $duracion = $horas."h y ".$minutos."min";
    $Partel = "";
    if($cmd == "RRA") $Partel = " - .numtodia($dia).",
".substr($fecha,8,2)."/".substr($fecha,5,2)."/".substr($fecha,0,4)." a las
".substr($hora, 0, 5)." \n          ·Regar lado A\n          ·Duración: ".$duracion."\n";
    if($cmd == "RRB") $Partel = " - .numtodia($dia).",
".substr($fecha,8,2)."/".substr($fecha,5,2)."/".substr($fecha,0,4)." a las
".substr($hora, 0, 5)." \n          ·Regar lado B\n          ·Duración: ".$duracion."\n";
    if($cmd == "RAB") $Partel = " - .numtodia($dia).",
".substr($fecha,8,2)."/".substr($fecha,5,2)."/".substr($fecha,0,4)." a las
".substr($hora, 0, 5)." \n          ·Regar todo\n          ·Duración: ".$duracion."\n";
    if($cmd == "DEP") $Partel = " - .numtodia($dia).",
".substr($fecha,8,2)."/".substr($fecha,5,2)."/".substr($fecha,0,4)." a las
".substr($hora, 0, 5)." \n          ·Depurar\n          ·Duración: ".$duracion."\n";
    return $Partel;
}

function numtodia($num){
    $dias = array('Domingo', 'Lunes', 'Martes', 'Miércoles', 'Jueves',
'Viernes', 'Sábado', 'Domingo');
    return $dias[$num];
}

// Contraseña -----
if ($_GET['pwd'] != $pwd){
    echo '<p>Acceso denegado</p>';
    die();
}

$Q1 = "SELECT * FROM Estado ORDER BY Id DESC";
$Estados = mysqli_query($con, $Q1);
$Estado = mysqli_fetch_array($Estados);

```

```

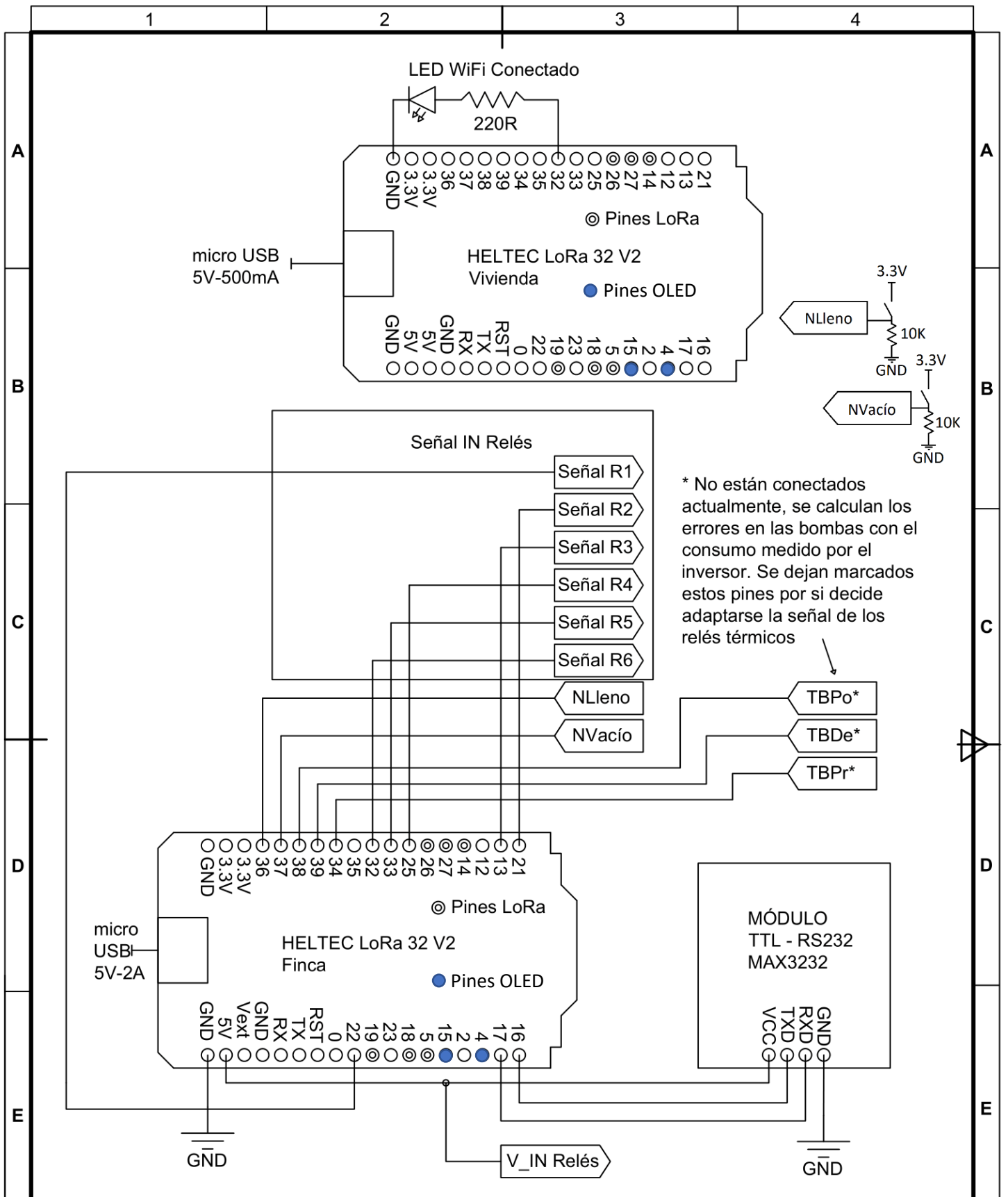
    if ($_GET['TASK'] == "Riego"){
        $Condition = "(Comando LIKE 'RRA%' OR Comando LIKE 'RRB%' OR Comando
LIKE 'RAB%')";
        if ($Estado["RegandoA"] == 1 && ($Estado["RegandoB"] == 1)) $Cabecera
= "Se está regando todo";
        else if ($Estado["RegandoA"] == 1) $Cabecera = "Se está regando el
lado A";
        else if ($Estado["RegandoB"] == 1) $Cabecera = "Se está regando el
lado B";
        else $Cabecera = "No se está regando";
        if ($Estado["WordSensores"] & 64) $Cabecera = "ALARMA TÉRMICO BOMBA
POZO";
        if ($Estado["ErrorCOMLoRa"] == 1) $Cabecera = "Esperando comunicación
LoRa...";
    }
    else if ($_GET['TASK'] == "Depuradora"){
        $Condition = "Comando LIKE 'DEP%'";
        if ($Estado["Depurando"] == 1) $Cabecera = "Se está depurando";
        else $Cabecera = "No se está depurando";
        if ($Estado["WordSensores"] & 128) $Cabecera = "ALARMA TÉRMICO
DEPURADORA";
        if ($Estado["ErrorCOMLoRa"] == 1) $Cabecera = "Esperando comunicación
LoRa...";
    }
    else{
        $Condition = "";
    }
    //Texto de tareas repetitivas
    $Q1 = "SELECT * FROM TareasCiclicas WHERE Dia > '0' AND Activo = '1' AND
$Condition ORDER BY Dia, Hora ASC";
    $Tareas = mysqli_query($con, $Q1);
    $TareasRepe = "Programas repetitivos: \n";
    while($NuevaTC = mysqli_fetch_array($Tareas)){
        $TareasRepe = $TareasRepe."\n".LeeComandoDiaHora($NuevaTC["Comando"],
$NuevaTC["Dia"], $NuevaTC["Hora"]);
    }
    $Q1 = "SELECT * FROM TareasCiclicas WHERE Dia = '0' AND Activo = '1' AND
$Condition ORDER BY Hora ASC";
    $Tareas = mysqli_query($con, $Q1);
    while($NuevaTC = mysqli_fetch_array($Tareas)){
        $TareasRepe = $TareasRepe."\n".LeeComandoDiaHora($NuevaTC["Comando"],
0, $NuevaTC["Hora"]);
    }
    //Texto de tareas únicas
    $TareasRepe = $TareasRepe."\n\n"."Programas únicos:\n";
    $fecha = getdate();
    $HoraActual = $fecha["hours"].":".$fecha["minutes"].":".$fecha["seconds"];
    $fehcaAct = date("Y-m-d");
    $fechalw = date("Y-m-d", strtotime($fehcaAct."+ 8days"));
    //$fehcaAct = $fehcaAct." ".$HoraActual;
    $fehcaAct = $fehcaAct." "." 00:00:00";
    $fechalw = $fechalw." 00:00:00";
    $Q1 = "SELECT * FROM TareasCola WHERE PendienteEnvio = '1' AND FechaHora >=
'$fehcaAct' AND FechaHora < '$fechalw' AND $Condition ORDER BY FechaHora ASC";
    $Tareas = mysqli_query($con, $Q1);
    while($NuevaTC = mysqli_fetch_array($Tareas)){
        $TareasRepe =
$TareasRepe."\n".LeeComandoDiaHora2($NuevaTC["Comando"], date('N',
strtotime(substr($NuevaTC["FechaHora"],0,10))), substr($NuevaTC["FechaHora"],-8,5),
$NuevaTC["FechaHora"]);
    }
    $TareasRepe = $TareasRepe."\n\n\n\n\n\n\n";
    $datajson = json_encode(['Tareas'=> $TareasRepe, 'Cabecera' => $Cabecera]);
    echo $datajson;
?>

```

CAPÍTULO 3. PLANOS

ÍNDICE DE PLANOS

ÍNDICE DE PLANOS	241
3.1. Esquema WiFi LoRa 32 (V2).....	242
3.2. Esquema de control	243
3.3. Esquema de potencia.....	244



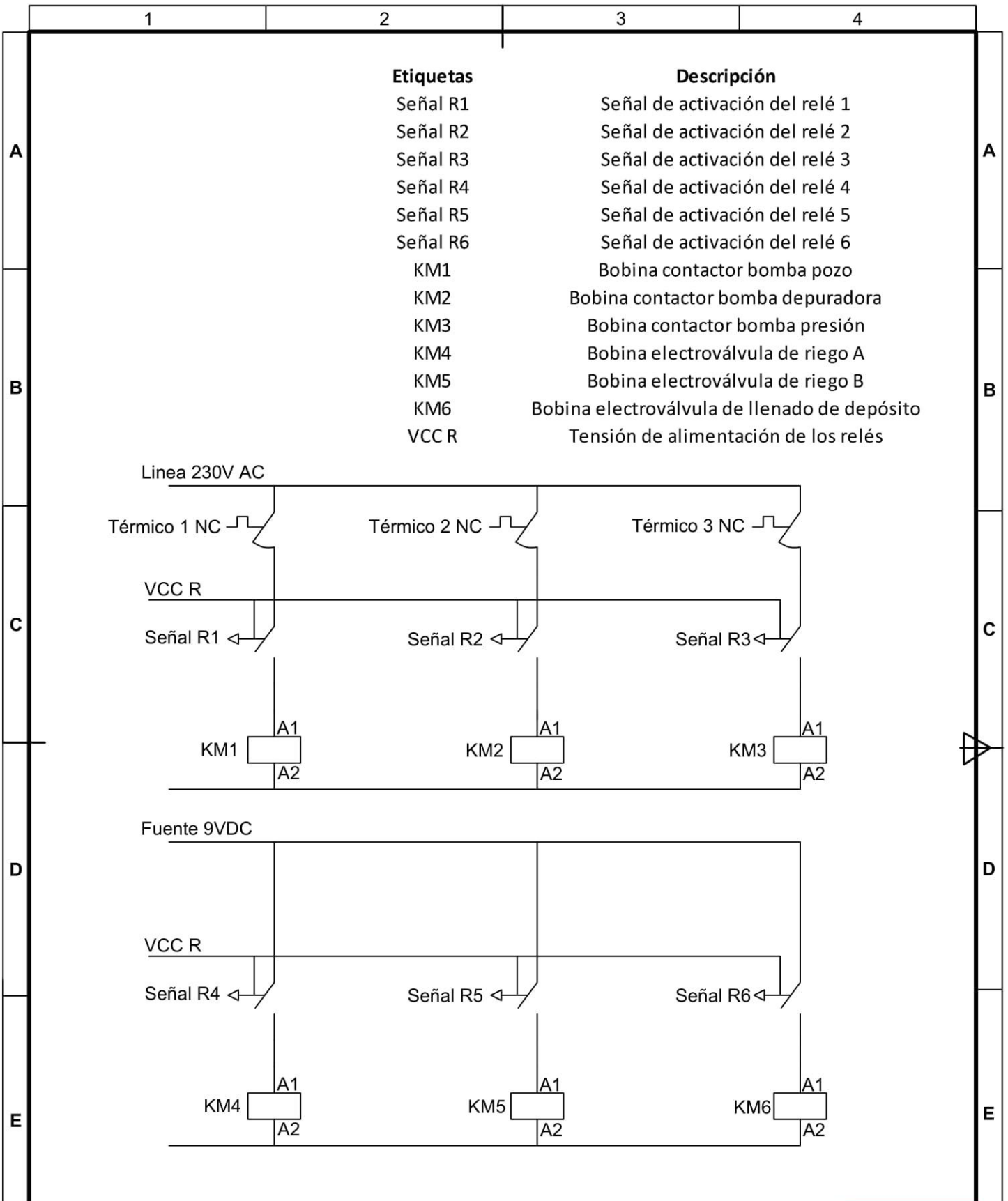
* No están conectados actualmente, se calculan los errores en las bombas con el consumo medido por el inversor. Se dejan marcados estos pines por si decide adaptarse la señal de los relés térmicos



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL
Universidad de La Rioja
Grado en Ingeniería Electrónica

	FECHA	NOMBRE
Dibujado	18-08-2019	Alejandro Navarro
Comprobado		
D.S.Normas	U.N.E	Tolerancia General

F	Escalas	ESQUEMA WiFi LoRa 32 (V2)	NÚMERO: 001
	Proyección	Control y monitorización de una finca con IoT y LPWAN	Referencia:
			Sustituye a
			Sustituido por

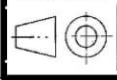


	FECHA	NOMBRE
Dibujado	18-08-2019	Alejandro Navarro
Comprobado		
D.S.Normas	U.N.E	Tolerancia General

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL
Universidad de La Rioja
Grado en Ingeniería Electrónica



F	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">Escalas</td> <td style="text-align: center;">ESQUEMA DE CONTROL</td> <td style="text-align: center;">NÚMERO: 002</td> </tr> <tr> <td style="text-align: center;">Proyección</td> <td style="text-align: center;">Control y monitorización de una finca con IoT y LPWAN</td> <td>Referencia:</td> </tr> <tr> <td></td> <td></td> <td>Sustituye a</td> </tr> <tr> <td></td> <td></td> <td>Sustituido por</td> </tr> </table>	Escalas	ESQUEMA DE CONTROL	NÚMERO: 002	Proyección	Control y monitorización de una finca con IoT y LPWAN	Referencia:			Sustituye a			Sustituido por	F
Escalas	ESQUEMA DE CONTROL	NÚMERO: 002												
Proyección	Control y monitorización de una finca con IoT y LPWAN	Referencia:												
		Sustituye a												
		Sustituido por												



1

2

3

4

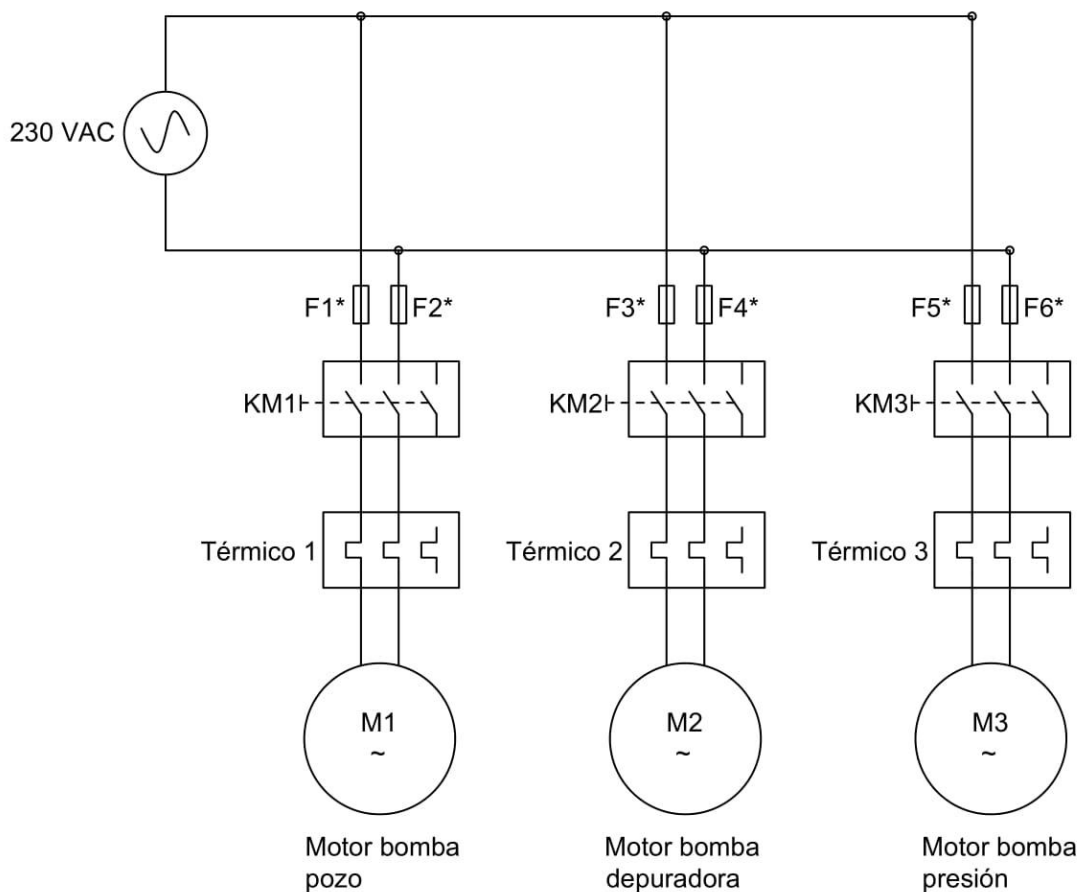
A

A

*Los fusibles ya están instalados,
por lo que no será necesario incluir
nuevos

B

B



C


C

D

D

E

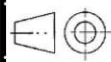
E

	FECHA	NOMBRE	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <i>Universidad de La Rioja</i> Grado en Ingeniería Electrónica	
Dibujado	18-08-2019	Alejandro Navarro		
Comprobado				
D.S.Normas	U.N.E	Tolerancia General		

F

F

Escalas	ESQUEMA DE POTENCIA	NÚMERO: 003
Proyección	Control y monitorización de una finca con IoT y LPWAN	Referencia:
		Sustituye a
		Sustituido por



CAPÍTULO 4. PLIEGO DE CONDICIONES

ÍNDICE PLIEGO DE CONDICIONES

ÍNDICE PLIEGO DE CONDICIONES.....	245
ÍNDICE DE ILUSTRACIONES.....	246
ÍNDICE DE TABLAS.....	246
4.1. Objeto del pliego de condiciones.....	247
4.2. Condiciones técnicas.....	247
4.2.1. Descripción de las nuevas instalaciones.....	247
4.2.2. Especificaciones de los elementos constitutivos del proyecto.....	250
4.2.2.1. Especificaciones de los elementos de la modificación en la instalación eléctrica	250
4.2.2.2. Especificaciones de los elementos de la modificación de la instalación hidráulica	250
4.2.2.3. Especificaciones del hardware de control y comunicaciones.....	251
4.2.2.4. Especificaciones Sensor de nivel.....	252
4.2.2.5. Especificaciones del protocolo LoRa desarrollado.....	253
4.2.2.6. Especificaciones del servidor instalado.....	253
4.2.3. Condiciones y comprobación de funcionamiento.....	255
4.2.3.1. Zona vivienda.....	255
4.2.3.2. Zona finca.....	256
4.3. Ejecución.....	258
4.4. Condiciones económicas.....	258
4.4.1. Garantía de funcionamiento.....	258
4.4.2. Impuestos.....	258
4.4.3. Actualización de precios.....	259
4.4.4. Reclamación por un aumento de precios.....	259
4.4.5. Retraso en los pagos.....	259
4.4.6. Retraso en los trabajos.....	259
4.5. Condiciones legales.....	260
4.5.1. Marco jurídico.....	260
4.5.2. Accidentes y daños a terceros.....	260
4.5.3. Responsabilidad civil.....	260

4.5.4.	Permisos	260
4.5.5.	Rescisión del contrato	261
4.5.6.	Normas y legislación aplicadas.....	261
4.5.6.1.	Legislación bandas de frecuencia ISM.....	261
4.5.6.2.	Normativa general para la industria.....	261

ÍNDICE DE ILUSTRACIONES

Ilustración 4. 1.	Tarjeta WiFi LoRa 32 (V2)	251
Ilustración 4. 2.	Módulo TTL-RS232	252
Ilustración 4. 3.	Sensor de nivel	252
Ilustración 4. 4.	Raspberry Pi Model 3B+	254
Ilustración 4. 5.	Mensaje 1 Tarjeta vivienda	255
Ilustración 4. 6.	Mensaje 2 Tarjeta vivienda	255
Ilustración 4. 7.	Mensaje 3 Tarjeta vivienda	255
Ilustración 4. 8.	Mensaje 4 Tarjeta vivienda	256
Ilustración 4. 9.	Mensaje 2 Tarjeta vivienda. In time!	256
Ilustración 4. 10.	Pantalla Finca sin error	256
Ilustración 4. 11.	Pantalla finca con error	257
Ilustración 4. 12.	Pantalla finca con última orden recibida.....	257
Ilustración 4. 13.	Mensaje pantalla finca. Enviando LoRa.....	257

ÍNDICE DE TABLAS

Tabla 4. 1.	Especificaciones componentes instalación eléctrica.....	250
Tabla 4. 2.	Especificaciones elementos instalación hidráulica.....	250
Tabla 4. 3.	Características técnicas Tarjeta WiFi LoRa 32 (V2).....	251
Tabla 4. 4.	Características eléctricas Tarjeta WiFi LoRa 32 (V2).....	251
Tabla 4. 5.	Características Módulo TTL-RS232	252
Tabla 4. 6.	Características sensor de nivel	252
Tabla 4. 7.	Características raspberry Pi Model 3B+.....	253

4.1. Objeto del pliego de condiciones

En este pliego se reúnen las condiciones y requisitos que se exigen al contratista para la correcta realización de proyecto en cuestión, “Control y monitorización de una finca con IoT y LPWAN”.

Se especifican las condiciones técnicas, las fechas de ejecución, las condiciones económicas y las condiciones legales del proyecto.

4.2. Condiciones técnicas

4.2.1. Descripción de las nuevas instalaciones

Las instalaciones que se van a realizar pretenden ser capaces de controlar y monitorizar una finca con los siguientes elementos:

- Generación de energía para autoconsumo
 - Partes constitutivas
 - Producción de energía solar fotovoltaica independiente de la red eléctrica.
 - Acumuladores para disponer de energía cuando no hay sol.
 - Grupo electrógeno diésel de emergencia para suplir la energía en caso de no disponer de sol ni batería.
 - Pretensiones:
 - Monitorizar la producción solar
 - Monitorizar el consumo de energía
 - Monitorizar el nivel de carga de los acumuladores

- Piscina
 - Partes constitutivas:
 - Depuradora
 - Pretensiones:
 - Controlar de forma automática y/o remota el accionamiento

- Sistema de riego.
 - Partes constitutivas:
 - Grifo de riego en zona A
 - Grifo de riego en zona B
 - Pretensiones
 - Accionar los grifos de forma remota y/o automática

- Gestión del agua:
 - Partes constitutivas:
 - Depósito de agua
 - Bomba de presión para abastecer a la casilla de la finca desde el depósito
 - Pozo con bomba de presión de agua para abastecer al depósito y al riego.
 - Pretensiones:
 - Controlar el nivel de agua del depósito
 - Gestionar el llenado del depósito
 - Gestionar la bomba de presión de agua que abastece la casilla de la finca desde el depósito.
 - Programar y controlar la bomba de presión del pozo para poder llevar un control de la cantidad de agua que se extrae en función del tiempo de activación de su bomba consultando la base de datos.

Para llevar a cabo el proyecto es necesario realizar una modificación en la instalación eléctrica e hidráulica.

En cuanto a la modificación en la instalación eléctrica, se ha añadido lo siguiente:

- Mejora en la seguridad. Se han añadido relés térmicos en las líneas de las bombas, que cortarían la corriente si se excede la corriente de trabajo configurada.
 - Características de los relés térmicos
 - Relé térmico para bomba más potente:
 - Consumo teórico: 7A
 - Regulación del relé: 7 a 10A
 - Relé térmico para las otras dos bombas
 - Consumos teóricos: 4A una y 5A otra
 - Regulación del relé: 4 a 6ª
- Control de las bombas con el microcontrolador elegido para el proyecto:
 - Contactores:
 - Función: dar potencia a las bombas
 - Tensión de control: 230VAC
 - Potencia soportada: 2,2kW
 - Un contactor por bomba
 - Relés de control:
 - Función: dar señal de control a los contactores
 - Tensión de alimentación: 5VDC
 - Tensión de control: 3.3V-5VDC
 - Salida: Hasta 10A con 250VAC-30VDC

Modificación en la instalación hidráulica:

- Electroválvulas para controlar el paso de agua al riego y al depósito.
 - Caudal teórico a manijas 7000 l/h
 - Caudal soportado: hasta 9000 l/h
 - Tensión de funcionamiento 9V
 - Relé de control:
 - Tensión de alimentación: 5VDC
 - Tensión de control: 3.3V-5VDC
 - Salida: Hasta 10A-250VAC-30VDC

Instalación de control:

- Tarjeta WiFi LoRa 32 V2 de Heltec:
 - 80MHz
 - LoRa
 - WiFi
 - Bluetooth
- Módulo TTL a RS232
 - Chip MAX3232
- Relés de control utilizados para los apartados anteriores

4.2.2. Especificaciones de los elementos constitutivos del proyecto

4.2.2.1. Especificaciones de los elementos de la modificación en la instalación eléctrica

Elemento	Descripción	Características
LRD14	Relé de protección térmica para proteger la bomba de mayor consumo	Protección: 7 a 10 A
		Tensión: Hasta 690 VAC
		Frecuencia: 0 a 400Hz
		Temperatura de operación: -20 a 60°C
LRD12	Relé de protección térmica para proteger la bomba intermedia	Protección: 5.5 a 8 A
		Tensión: Hasta 690 VAC
		Frecuencia: 0 a 400Hz
		Temperatura de operación: -20 a 60°C
LRD10	Relé de protección térmica para proteger la bomba de menor consumo	Protección: 4 a 6 A
		Tensión: Hasta 690 VAC
		Frecuencia: 0 a 400Hz
		Temperatura de operación: -20 a 60°C
LC1D09P7	Contactor para activar las bombas	Potencia soportada a 230V: 2,2kW
		Potencia máxima: 5,5kW a 500-600VAC
		Temperatura de operación: -20 a 60°C
		Especificación AC-3
		Tensión nominal de la bobina: 230VAC
Single Relay Board	Módulo relé para el control de contactores y electroválvulas	Cargas hasta 10A
		Cargas hasta 250VAC o 30VDC
		Aislamiento entre microcontrolador y carga
		Alimentación: 5VDC
		Señal de control: 3.3VDC-5VDC
		Temperatura de operación: -25 a 70°C
Transformador 9V	Fuente de 9V para el control de las electroválvulas	Tensión de entrada: 230 VAC
		Tensión de salida: 9VDC
		Carga máxima salida: 1A

Tabla 4. 1. Especificaciones componentes instalación eléctrica

4.2.2.2. Especificaciones de los elementos de la modificación de la instalación hidráulica

Elemento	Descripción	Características
Hunter PGV-100G-B	Electroválvula para el control del flujo de agua para riego y llenado de depósito	Tensión de la bobina: 9V
		Medida: 1"
		Caudal mínimo: 0,7 l/min
		Caudal máximo 150 l/min

Tabla 4. 2. Especificaciones elementos instalación hidráulica

4.2.2.3. Especificaciones del hardware de control y comunicaciones

- Tarjeta WiFi LoRa 32 V2. Heltec.

CARACTERÍSTICAS TÉCNICAS	
Parámetro	Características
MCU	ESP32(240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth)
Chip LoRa	SX1276
Bandas LoRa	EU_433, CN_470_510, EU_863_870, US_902_928 cuatro bandas opcionales
Máxima potencia LoRa	18dB ± 2dB
Recursos de hardware	3x UART, 2x SPI, 2x IDC, 1x I2S
	12 bits ADC - 8 bits DAC
	29 GPIO
Pantalla	OLED 0,96"
Memoria flash	8MB SPI FLASH
Interfaz	MicroUSB, Antena LoRa

Tabla 4. 3. Características técnicas Tarjeta WiFi LoRa 32 (V2)

CARACTERÍSTICAS ELÉCTRICAS	
Parámetro	Características
Alimentación	USB (≥500mA) Min:4.7V - Típico: 5V - Max: 6V
	Batería de litio (≥250mA) Min:3.3V - Típico: 3.7V - Max: 4.2V
	3.3V (pin) powered (≥150mA) Min:2.7V - Típico: 3.3V - Max: 3.5V
	5V (pin) powered (≥500mA) Min:4.7V - Típico: 5V - Max: 6V
Consumos	WiFi scan 115mA
	WiFi AP 135mA
	LoRa 10dB emission 50mA
	LoRa 12dB emission 60mA
	LoRa 15dB emission 110mA
	LoRa 20dB emission 130mA
	Máximo 500MAh a la salida del pin de 3.3V
	Salida del pin de 5V = a la salida de la alimentación

Tabla 4. 4. Características eléctricas Tarjeta WiFi LoRa 32 (V2)

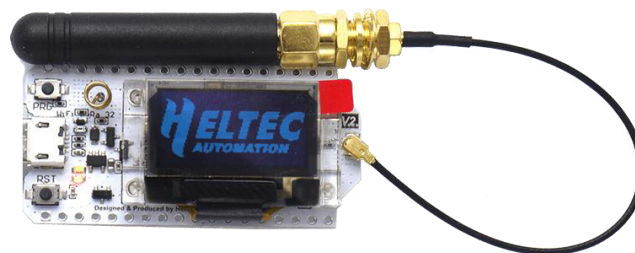


Ilustración 4. 1. Tarjeta WiFi LoRa 32 (V2)

- Módulo TTL -RS232.

Parámetro	Características
Chip	MAX3232
Tensión de alimentación	3.3V - 5V
Tensión de lógica	3.3V - 5V
Máxima velocidad de comunicación	250kbps
Entradas	Pines Rx, Tx, GND, VCC
Salidas	Conector hembra db9 RS232

Tabla 4. 5. Características Módulo TTL-RS232

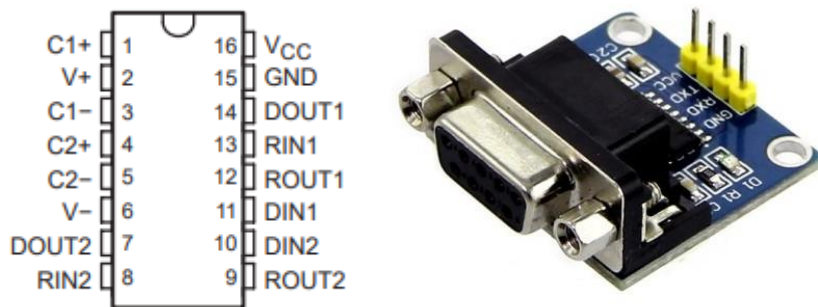


Ilustración 4. 2. Módulo TTL-RS232

4.2.2.4. Especificaciones Sensor de nivel

Parámetro	Características
Salida	Interruptor flotante simple
Tipo	Líquido
Corriente	1A
Densidad relativa mínima del líquido	0,85
Voltaje nominal	300VAC, 300VDC

Tabla 4. 6. Características sensor de nivel



Ilustración 4. 3. Sensor de nivel

4.2.2.5. Especificaciones del protocolo LoRa desarrollado

El protocolo LoRa desarrollado cumple la legislación europea sobre bandas de frecuencia ISM (industriales, científicas y médicas).

De modo que deberá tenerse en cuenta que para los cálculos realizados en anexos y aplicados en la memoria, no deberán sobrepasarse los 1866 mensajes diarios.

Este punto queda garantizado por la parte de comunicaciones que no depende del usuario, como es el envío de datos desde la finca a la vivienda.

Sin embargo, limita el número de órdenes que el usuario puede enviar desde la interfaz de usuario a los 1866 mensajes por nodo.

No debería suponer ninguna limitación de funcionamiento ya que esto implica un uso desorbitado del sistema.

Con un uso habitual es prácticamente imposible pasar de los 20 mensajes diarios por parte del usuario.

El margen es tan grande que no se ha limitado a nivel de software el uso de la aplicación para alcanzar este límite.

De todas formas, el usuario debe saber que para cumplir la normativa no debe superar este número de mensajes.

4.2.2.6. Especificaciones del servidor instalado

El servidor está instalado sobre una Raspberry Pi Modelo 3B+ que cuenta con las siguientes características:

CARACTERÍSTICAS RASPBERRY PI MODELO 3B+	
CPU + GPU	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
RAM	1GB LPDDR2 SDRAM
Wi-Fi	2.4GHz y 5GHz IEEE 802.11.b/g/n/ac
Bluetooth	4.2
Ethernet	Gigabit ethernet sobre USB 2.0. Máximo 300Mbps
GPIO	40 pines
Almacenamiento	Tarjeta SD donde se aloja el sistema operativo (caso instalado: 32GB)
Otros	HDMI, 4 USB 2.0, Puerto CSI y DSI, Salida de audio

Tabla 4. 7. Características raspberry Pi Model 3B+

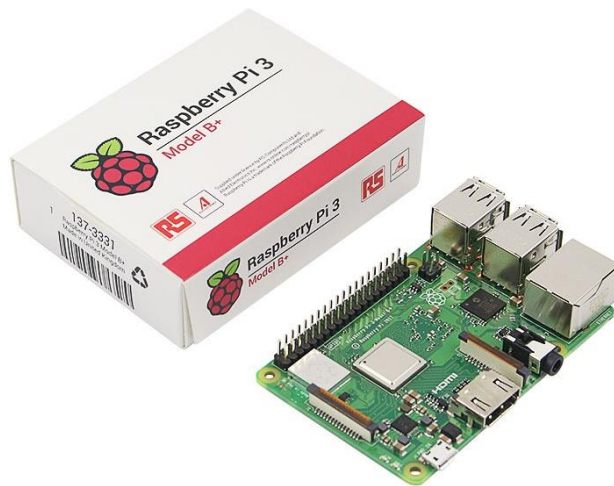


Ilustración 4. 4. Raspberry Pi Model 3B+

Además, este servidor está conectado a una red de fibra óptica simétrica de 300Mbps vía ethernet, por lo que la velocidad de conexión no debería ser un problema para el número de usuarios de la aplicación.

4.2.3. Condiciones y comprobación de funcionamiento

Para comprobar que el sistema está funcionando correctamente se puede observar su estado en las pantallas OLED de las tarjetas ESP32, que nos darán la siguiente información:

4.2.3.1. Zona vivienda

Al iniciar la tarjeta muestra el siguiente mensaje que indica que se está conectando a la red WiFi:



Ilustración 4.5. Mensaje 1 Tarjeta vivienda

Una vez conectado la información es la siguiente:

- Línea 1: Estado de la conexión
- Línea 2: Tiempo de espera desde último mensaje
- Línea 3: Esperando LoRa... o InTime! En función de la última recepción



Ilustración 4.6. Mensaje 2 Tarjeta vivienda

Si pasados 65 segundos no ha recibido nada muestra el error, y sigue esperando al primer mensaje LoRa:



Ilustración 4.7. Mensaje 3 Tarjeta vivienda

Cuando recibe un mensaje LoRa muestra la siguiente pantalla:



Ilustración 4. 8. Mensaje 4 Tarjeta vivienda

Y vuelve al mensaje 2, mostrando esta vez “InTime!” En lugar de “Esperando LoRa...” mientras el tiempo sea menor a 57s.

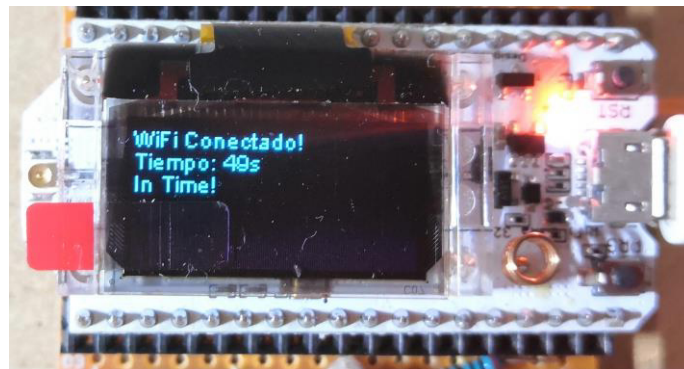


Ilustración 4. 9. Mensaje 2 Tarjeta vivienda. In time!

4.2.3.2. Zona finca

La pantalla en esta tarjeta muestra 3 líneas:

- Porcentaje de tiempo hasta el próximo mensaje.
- Última orden. Muestra cual ha sido la última orden recibida. (Ayuda a saber si hay un fallo en el programa o en las comunicaciones)
- Error inversor. Muestra un mensaje de error inversor si no recibe comunicaciones de éste. En caso de funcionar bien esta línea está vacía.

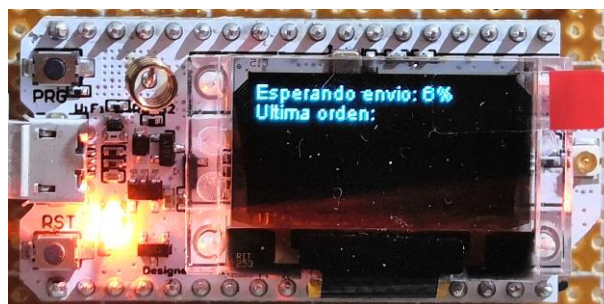


Ilustración 4. 10. Pantalla Finca sin error

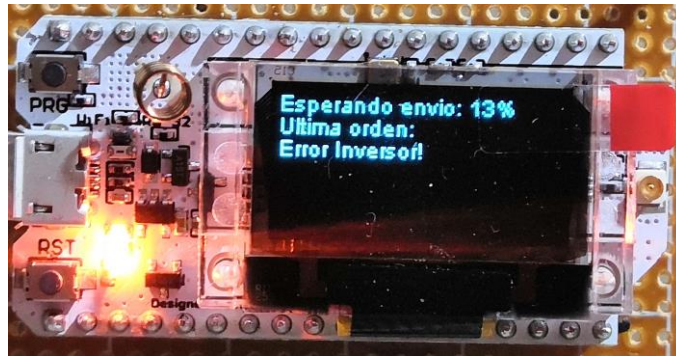


Ilustración 4. 11. Pantalla finca con error

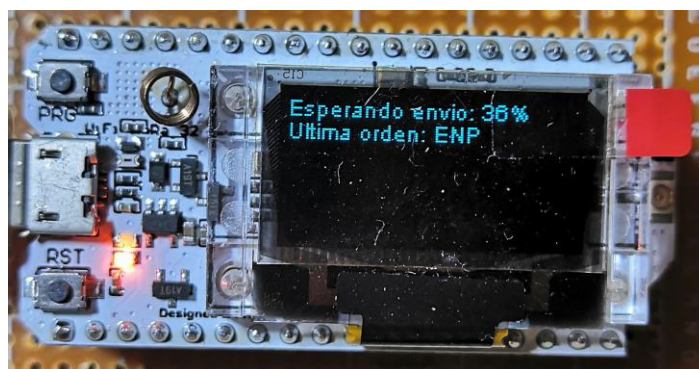


Ilustración 4. 12. Pantalla finca con última orden recibida

Además, muestra este mensaje cuando envía un nuevo dato:



Ilustración 4. 13. Mensaje pantalla finca. Enviando LoRa

4.3. Ejecución

En primer lugar, se terminará la instalación el viernes, 20 de septiembre de 2019, comenzando un periodo de evaluación de 1 mes, en el que se realizarán las pruebas y mejoras necesarias para la puesta a punto del sistema.

El lunes, 21 de octubre de 2019 se dará por finalizado el periodo de puesta a punto, pasando al periodo de garantía de funcionamiento del sistema.

4.4. Condiciones económicas

4.4.1. Garantía de funcionamiento

Durante el periodo de garantía, el adjudicatario debe garantizar los bienes del objeto del presente contrato, además de la instalación realizada, ofreciendo el soporte técnico y cambios precisos para evitar deficiencias de funcionamiento del sistema causadas por un elemento defectuoso o un mal funcionamiento de éste.

El periodo de garantía de funcionamiento comenzará con la finalización de la etapa de puesta a punto. Durante el periodo de garantía, ésta comprenderá:

- Reparación o sustitución de elementos defectuosos o dañados por un mal funcionamiento del sistema
- Mantenimiento del funcionamiento del sistema
- Actualizaciones y mejoras del sistema

Una vez finalizado el periodo de garantía, se deberá seguir manteniendo el funcionamiento del servidor de manera gratuita durante al menos 5 años desde la finalización.

4.4.2. Impuestos

Los impuestos derivados de la ejecución del proyecto por disposición de cualquier entidad pública (municipio, provincia, comunidad autónoma o estado) deberán ser abonados por el contratista, a excepción del IVA.

Las modificaciones en este aspecto tras el contrato únicamente afectarán al sujeto pasivo. Ninguna de las partes podrá apelar a estos cambios como causa de una revisión en los precios.

4.4.3. Actualización de precios

Si ocurre un caso en virtud del cual es necesario establecer un nuevo precio, será estudiado y acordado de la siguiente manera:

- El adjudicatario formulará por escrito, bajo su firma, el que considere que debe aplicarse.
- La Dirección Técnica estudiará el nuevo precio que considere que debe aplicarse.

Si ambos son coincidentes, se redacta un acta para formalizar el cambio de precios y en caso de no llegar a un punto de acuerdo, será el propietario el que tome la resolución que estime oportuna.

Estos cambios solo podrán aplicarse a unidades de obra que al momento de esta resolución no hubieran comenzado, de otra manera, el adjudicatario está obligado a aceptar el que desee fijar el responsable.

4.4.4. Reclamación por un aumento de precios

En caso de que el contratista, con anterioridad a la firma del contrato no hubiese llevado a cabo la reclamación oportuna, no podrá reclamar aumento de los precios establecidos en el presupuesto de base para la ejecución del proyecto, bajo ningún pretexto de error u omisión.

Los errores materiales o aritméticos en las unidades del importe se corregirán en cualquier momento en el que se constaten. No se tendrán en cuenta a los efectos de la rescisión de contrato.

4.4.5. Retraso en los pagos

Por ningún motivo de retraso en los pagos, el Contratista podrá suspender los trabajos, ni ejecutarlos a menor ritmo del que corresponda, de acuerdo con el plazo en el cual deben terminarse.

4.4.6. Retraso en los trabajos

EN caso de retraso no justificado, el contratista debe abonar una indemnización. El importe de ésta será la suma de perjuicios materiales a cauda de la imposibilidad de ocupación del terreno. Los daños deberán estar debidamente justificados.

4.5. Condiciones legales

4.5.1. Marco jurídico

La ejecución del proyecto deberá tener en cuenta las leyes vigentes en España y la comunidad europea.

Es responsabilidad de la empresa la ejecución del proyecto bajo las condiciones establecidas en el contrato y en los documentos que forman el proyecto, así como por las establecidas en las leyes de contratos de trabajo.

4.5.2. Accidentes y daños a terceros

El máximo responsable, y el único, de la seguridad de la fabricación es el contratista, que tiene la obligación de comprobar y cumplir el cumplimiento de las leyes de seguridad laboral. Por su parte, la propiedad queda libre de responsabilidades.

Además, es el contratista el encargado de establecer las medidas de seguridad necesarias, sí como de suministrar el material de seguridad necesario a los trabajadores.

En caso de accidente o daños a terceras personas u objetos debido a un incumplimiento de las leyes vigentes, el responsable será el contratista.

4.5.3. Responsabilidad civil

El contratista deberá tener cubierta su responsabilidad civil, a su vez, será el encargado de tomar las medidas oportunas para no comprometer la salud empleados ni terceras personas, siendo responsable en caso de accidente. En caso de accidente será el único responsable de la indemnización. Quedan exentos de responsabilidades el propietario y dirección facultativa.

4.5.4. Permisos

El contratista será el encargado del cumplimiento de las leyes para la realización del proyecto, cayendo sobre él la responsabilidad en caso de incumplimiento. Quedará exento por un mal uso de las instalaciones realizadas por parte del usuario indicado en las condiciones técnicas.

El contratista deberá encargarse de tener todos los permisos y certificados necesarios para la realización del proyecto, pero los costes irán a cargo de la propiedad.

4.5.5. Rescisión del contrato

Lo siguientes motivos serán las causantes de una rescisión del contrato:

- Comenzar con retraso con respecto a la fecha estipulada.
- Demora excesiva en los plazos de entrega.
- Cambios en el contrato que impliquen subidas de precio mayores al 8%.
- Incumplimiento de responsabilidades por parte del contratista.
- Fallos en los materiales a causa de negligencia por parte del contratista.

4.5.6. Normas y legislación aplicadas

4.5.6.1. *Legislación bandas de frecuencia ISM*

Orden IET/787/2013, de 25 de abril, por la que se aprueba el cuadro nacional de atribución de frecuencias.

UN - 39 Banda 868-870 MHz

Dispositivos de baja potencia no específicos para aplicaciones genéricas:

- 868,000 - 868,600 MHz con 25 mW (p.r.a.) de potencia radiada aparente máxima. Estos dispositivos deberán utilizar técnicas de acceso y mitigación de interferencias con rendimiento al menos equivalente a las técnicas descritas en las normas armonizadas según la Directiva 1999/5/CE, o alternativamente no sobrepasar el 1% de ciclo de trabajo. Se excluyen las aplicaciones analógicas de video.

4.5.6.2. *Normativa general para la industria*

Convenio 155 de la OIT sobre seguridad y salud de los trabajadores, de 22 de junio de 1981.

Real Decreto 707/2002, de 19 de julio, por el que se aprueba el Reglamento sobre el procedimiento administrativo especial de actuación de la Inspección de Trabajo y Seguridad Social y para la imposición de las medidas correctoras de incumplimientos en materia de prevención de riesgos laborales en el ámbito de la Administración General del Estado.

Ley 21/1992, de 16 de julio, de Industria.

Real Decreto 825/1993, de 28 de mayo, que determina Medidas Laborales y de Seguridad Social específicas a las que se refiere el artículo 6 de la Ley 21/1992.

Orden de 29 de julio de 1993, que desarrolla el Real Decreto 825/1993.

Real Decreto 697/1995, de 28 de abril, que desarrolla el Reglamento de Registro de Establecimientos Industriales de la Ley 21/1992.

Real Decreto 2526/1998, de 27 de noviembre, que modifica el artículo 17.1 del anexo al Real Decreto 697/1995.

Real Decreto A1823/1998, de 28 de agosto, sobre Composición y Funcionamiento de la Comisión para la Competitividad industrial, desarrollando la Ley 21/1992.

Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales.

Ley 54/2003, de 12 de noviembre, de reforma del marco normativo de la Prevención de Riesgos Laborales.

Real Decreto 171/2004, de 30 de enero, por el que se desarrolla el artículo 24 de la Ley 31/1995 de Prevención de Riesgos Laborales.

Directiva 89/391/CEE, relativa a la aplicación de las medidas para promover la mejora en la seguridad y de la salud de los trabajadores en el trabajo.

Ley 50/1998, de 30 de diciembre, de Medidas Fiscales, Administrativas y del Orden Social, que modifica la Ley 31/1995 en sus artículos 47, 48 y 49.

Real Decreto Legislativo 5/2000, de 4 de agosto, por el que se aprueba el texto refundido de la Ley sobre Infracciones y Sanciones en el Orden Social.

CAPÍTULO 5. MEDICIONES

ÍNDICE DE MEDICIONES

ÍNDICE DE MEDICIONES	263
5.1. Elementos materiales.....	264
5.2. Mano de obra.....	267

5.1. Elementos materiales

Partidas	Partes por unidad	Descripción	Cantidad
	WiFi LoRa 32 (V2) Heltec	Tarjeta de control y comunicaciones	2
	Módulo MAX3232	Módulo adaptador TTL a RS232	1
Control del sistema			
	Conector hembra RJ45	Conector hembra RJ45	1
	Módulo Relé 3.3V-5V	Módulo relé control contactores y electroválvulas	6
	LC1D09P7	Contactador de control de bombas	3
	LRD14	Relé térmico 7 a 10 A	1
Elementos de potencia, protección y alimentación			
	LRD10	Relé térmico 4 a 6 A	2
	Transformador 9V	Alimentación de electroválvulas	1
	Transformador 5V microUSB	Alimentación WiFi LoRa 32 (V2)	2

Partidas	Partes por unidad	Descripción	Cantidad
Elementos hidráulicos	HUNTER PGV-100G-B	Electroválvulas a 9V - hasta 9000 l/h	3
	RaspBerry Pi Modelo 3 B+	Ordenador para el servidor	1
Servidor	Caja Raspberry Pi	Caja para montar RaspBerry Pi	1
	Fuente alimentación Raspberry Pi	Alimentación micro USB par Raspberry Pi	1
	Kingston SDCS/32GBSP	Tarjeta microSD de 32 gigabytes para el sistema operativo	1
Materiales maqueta	Tablero 200x300 mm	Tablero de madera para maqueta	2
	LEDs	Diodos LED para simular sensores y actuadores	11
	SWITCH tipo palanca	Interruptores para simular sensores	5

Partidas	Partes por unidad	Descripción	Cantidad
Instalación	Caja estanca 90x90mm	Cajas para módulos de comunicaciones y relés de control	3
	PCB prototipos	Tarjetas PCB agujereadas 70x50mm	2
	Pack pin header hembra y macho	Pack con cabezal para pines. Soporte para la tarjeta y los componentes.	1
	Pack de 20 terminales de conexión. 2 polos	Terminales para conectar salidas y entradas digitales	1
	Caja estanca carril din	Caja para elementos de potencia y protecciones. Espacio 10 módulos	1

5.2. Mano de obra

Para la realización del proyecto será necesaria mano de obra para cubrir los siguientes aspectos:

- Programación del servidor
- Programación de la aplicación
- Programación del control
- Instalación

Mano de obra	Actividad	Cantidad	Horas diarias	Días
Programador	Programación del servidor	1	8	5
Programador	Programación de la aplicación android	1	8	8
Programador	Programación del control	1	8	5
Técnico	Instalación del sistema	1	8	2

CAPÍTULO 6. PRESUPUESTO

ÍNDICE DE PRESUPUESTO

ÍNDICE DE PRESUPUESTO	268
6.1. Cuadro de precios unitarios	269
6.1.1. Elementos materiales.....	269
6.1.2. Mano de obra	270
6.2. Cuadro de precios unitarios de las unidades de obra	271
6.3. Presupuesto	273
6.3.1. Elementos materiales.....	273
6.3.2. Mano de obra	274
6.3.3. Impuestos y tasas	274
6.3.4. Presupuesto.....	275

6.1. Cuadro de precios unitarios

6.1.1. Elementos materiales

Elemento	Precio unitario SIN IVA	Precio unitario CON IVA
WiFi LoRa 32 (V2) Heltec	14,00 €	16,94 €
Módulo MAX3232 TTL - RS232	3,38 €	4,09 €
Sensor de nivel	17,75 €	21,48€
Conector hembra RJ45	2,27 €	2,75 €
Módulo Relé 3.3V-5V	1,78 €	2,15 €
LC1D09P7	28,00 €	33,88 €
LRD14	30,67 €	37,11 €
LRD12	30,67 €	37,11 €
LRD10	30,67 €	37,11 €
Transformador 9V	8,25 €	9,98 €
Transformador 5V microUSB	8,25 €	9,98 €
HUNTER PGV-100G-B	25,00 €	30,25 €
RaspBerry Pi Modelo 3 B+	30,00 €	36,30 €
Caja Raspberry Pi	4,09 €	4,95 €
Fuente alimentación Raspberry Pi	6,60 €	7,99 €
Kingston SDCS/32GBSP	4,95 €	5,99 €
Tablero 200x300 mm	0,48 €	0,58 €
LEDs	0,06 €	0,07 €
SWITCH tipo palanca	0,81 €	0,98 €
Caja estanca 90x90mm	1,64 €	1,98 €
Caja estanca carril din	10,70 €	12,95 €
PCB de prototipos	1,86 €	2,25 €
Cable de 3 hilos 1,5mm ² (Uds m)	0,72 €	0,87 €
Pack de terminales de conexión	5,46 €	6,61 €
Pack pin header macho y hembra	0,11 €	0,13 €

6.1.2. Mano de obra

Elemento	Precio unitario SIN IVA (€)	Precio unitario CON IVA
Programador servidor	30,00 €	36,30 €
Programador Android	30,00 €	36,30 €
Programador control	30,00 €	36,30 €
Técnico instalación	25,00 €	30,25 €

6.2. Cuadro de precios unitarios de las unidades de obra

Unidad de obra	Elemento	Precio unitario CON IVA	Cantidad	Precio	Precio unidad de obra con IVA
Control del sistema	WiFi LoRa 32 (V2) Heltec	16,94 €	2	33,88 €	96,60 €
	Módulo MAX3232 TTL - RS232	4,09 €	1	4,09 €	
	Sensor de nivel	21,48 €	2	42,96 €	
	Conector hembra RJ45	2,75 €	1	2,75 €	
	Módulo Relé 3.3V-5V	2,15 €	6	12,92 €	
Elementos de potencia, protección y alimentación	LC1D09P7	33,88 €	3	101,64 €	242,92 €
	LRD14	37,11 €	1	37,11 €	
	LRD12	37,11 €	1	37,11 €	
	LRD10	37,11 €	1	37,11 €	
	Transformador 9V	9,98 €	1	9,98 €	
	Transformador 5V micro USB	9,98 €	2	19,97 €	

Unidad de obra	Elemento	Precio unitario CON IVA	Cantidad	Precio	Precio unidad de obra con IVA
Elementos hidráulicos	HUNTER PGV-100G-B	30,25 €	3	90,75 €	90,75 €
Servidor	RaspBerry Pi Modelo 3 B+	36,30 €	1	36,30 €	55,22 €
	Caja RaspBerry Pi	4,95 €	1	4,95 €	
	Fuente alimentación RaspBerry Pi	7,99 €	1	7,99 €	
	Kingston SDCS/32GBSP	5,99 €	1	5,99 €	
Maqueta	Tablero 200x300 mm	0,58 €	2	1,16 €	6,86 €
	LEDs	0,07 €	11	0,80 €	
	SWITCH tipo palanca	0,98 €	5	4,90 €	
Instalación	Caja estanca 90x90mm	1,98 €	3	5,95 €	34,50 €
	Caja estanca carril din	12,95 €	1	12,95 €	
	PCB de prototipos	2,25 €	2	4,50 €	
	Cable de 3 hilos 1,5mm2 (Uds m)	0,87 €	5	4,36 €	
	Pack de terminales de conexión	6,61 €	1	6,61 €	
	Pack pin header macho y hembra	0,13 €	1	0,13 €	

6.3. Presupuesto

6.3.1. Elementos materiales

Unidad de obra	Precio unidad de obra con IVA	Cantidad	Precio con IVA
Control del sistema	96,60 €	1	96,60 €
Elementos de potencia, protección y alimentación	242,92 €	1	242,92 €
Elementos hidráulicos	90,75 €	1	90,75 €
Servidor	55,22 €	1	55,22 €
Maqueta	6,86 €	1	6,86 €
Instalación	34,50 €	1	34,50 €
	Precio total materiales con IVA		526,35 €

6.3.2. Mano de obra

Elemento	Precio unitario SIN IVA (€)	Precio unitario CON IVA	Horas/día	Días	Total
Programador servidor	30,00 €	36,30 €	8	5	1.452,00 €
Programador Android	30,00 €	36,30 €	8	8	2.323,20 €
Programador control	30,00 €	36,30 €	8	5	1.452,00 €
Técnico instalación	25,00 €	30,25 €	8	2	484,00 €
				Precio total mano de obra	5.711,20 €

6.3.3. Impuestos y tasas

No se han aplicado más impuestos ni tasas a parte del IVA ya incluido en los cuadros de precios.

6.3.4. Presupuesto

Elemento	Precio total con IVA
Materiales	526,35 €
Mano de obra	5.711,20 €
Gastos generales	250 €
Beneficio industrial (20%)	1.297,51 €
Total	7.785,06 €